



DETECCIÓN DE HUMO UTILIZANDO PROCESADO DE IMAGEN

(SMOKE DETECTION USING IMAGE PROCESSING)

Proyecto de fin de carrera

**Escola Tècnica d'Enginyeria de Telecomunicació de
Barcelona**

Universitat Politècnica de Catalunya

por

José María Granado Rodríguez

**En cumplimiento parcial de
los requisitos para el grado de
Ingeniería de Sistemas Audiovisuales**

Directora: Montse Pardàs Feliu

Barcelona, Julio 2018

Abstract

The objective of this final project of degree has been to design a smoke detection system in forest areas, following a pilot that was made in another project at the UPC in 2011.

The project has been divided into 3 stages: analysis of current techniques in smoke detection, implementation of techniques and analysis of results.

The implementation has been segmented into two blocks, one of them of background analysis (background subtraction) to detect the objects that move in the video, and another to classify these objects, using models of neural networks already trained and adapting them to this case.

For the classification, a database of manually collected images has been used, with a total of 1645 images for training and other 102 for the validation and adjustment of parameters.

Finally the results of the project have been quite successful since it was intended to detect and classify smoke in the videos and it has been achieved with an 88.22% accuracy for smoke and with a low number of false alarms of 6%

Resum

L'objectiu d'aquest projecte final de carrera ha consistit a dissenyar un sistema de detecció de fums en zones forestals continuant un pilot que es va fer en un altre projecte a la UPC el 2011.

El projecte s'ha dividit en 3 etapes: anàlisi de les tècniques actuals en detecció de fum, implementació de les tècniques i anàlisi dels resultats.

S'ha segmentat en dos blocs la implementació, un d'ells d'anàlisi del fons de les imatges (background subtraction) per detectar els objectes que es mouen en el vídeo, i un altre per a la classificació d'aquests objectes, utilitzant models de xarxes neuronals ja entrenats i adaptant-los a aquest cas.

Per a la classificació s'ha utilitzat una base de dades d'imatges recopilades manualment, amb un total de 1646 imatges per l'entrenament i 102 per a la validació i ajust dels paràmetres.

Finalment, els resultats del projecte han estat bastant acertats ja que es buscava detectar i classificar fum als vídeos i s'ha aconseguit amb un 88,22 % d'encert i amb un nombre baix de falses alarmes del 6%.

Resumen

El objetivo de este proyecto final de carrera ha consistido en diseñar un sistema de detección de humos en zonas forestales continuando un piloto que se hizo en otro proyecto en la UPC en el 2011.

El proyecto se ha dividido en 3 etapas: análisis de las técnicas actuales en detección de humo, implementación de las técnicas y análisis de los resultados.

Se ha segmentado en dos bloques la implementación, uno de ellos de análisis del fondo de las imágenes (background subtraction) para detectar los objetos que se mueven en el video, y otro para la clasificación de estos objetos, utilizando modelos de redes neuronales ya entrenados y adaptándolos a este caso.

Para la clasificación se ha utilizado una base de datos de imágenes recogidas manualmente, con un total de 1646 imágenes para el entrenamiento y otras 102 para la validación y ajuste de parámetros.

Finalmente los resultados del proyecto han sido bastante acertados ya que se buscaba detectar y clasificar humo en los videos y se ha conseguido con un 88,22% de acierto para el humo y con un número bajo de falsas alarmas del 6%



Agradecimientos

Quiero agradecer en primer lugar a mi directora de proyecto Montse Pardàs por darme la oportunidad de hacer este trabajo tan interesante y por su ayuda en todos los aspectos del proyecto.

Y también agradecer a mi familia y a mis amigos por el apoyo durante no solo el tiempo que he estado haciendo este trabajo de fin de grado sino por todo el apoyo que me han ofrecido durante los años en los que he cursado la carrera y que me han llevado hasta aquí

Historial de revisiones y registro de aprobación

Revisión	Fecha	Objetivo
0	01/06/2018	Creación del documento
1	17/06/2018	Revisión del documento
2	22/06/2018	Revisión del documento
3	04/07/2018	Revisión del documento
4	06/07/2018	Revisión final

LISTA DE DISTRIBUCIÓN

Name	e-mail
José María Granado Rodríguez	chema_13_1993@hotmail.com
Montse Pardàs	montse.pardas@upc.edu

Escrito por:		Revisado y aprobado por	
Fecha	01/06/2018	Fecha	06/07/2018
Nombre	José María Granado Rodríguez	Nombre	Montse Pardàs
Posición	Autor del proyecto	Posición	Directora del proyecto

Tabla de contenidos

1.	Introducción	11
1.1.	Objetivos del proyecto	11
1.2.	Requerimientos y especificaciones	11
1.3.	Procedimiento seguido:	11
1.4.	Plan de trabajo	12
1.5.	Desviaciones respecto al plan de trabajo	13
2.	Estado del arte de la tecnología aplicada en este trabajo	14
2.1.	Técnicas de detección de fondo y detección de movimiento	14
2.2.	Clasificación de las imágenes	17
3.	Metodología/Desarrollo del proyecto:	19
3.1	Esquema general	20
3.2.	Modelo de fondo utilizado	21
3.3.	Búsqueda de componentes conexas	25
3.4.	Clasificador de humo	26
3.5.	Analizador de salida del algoritmo	31
4.	Resultados	35
4.1	Análisis y comentarios de los resultados de cada video de la base de datos	35
4.2	Análisis de los resultados finales	48
5.	Budget	51
6.	Conclusiones y mejoras a futuro	52

Lista de Figuras

En la lista de a continuación están definidas todas las figuras que aparecen en el proyecto:

Figura 1. Diagrama de Gantt	13
Figura 2. Ejemplo de tres gaussianas	16
Figura 3. Esquema general del algoritmo	21
Figura 4. Elemento estructurante, elipse (5 – 5)	23
Figura 5. Ejemplo de apertura morfológica	24
Figura 6. Ejemplo de sustracción de fondo para un frame de algunos vídeos utilizados	25
Figura 7. Ejemplo de máscara binaria en la que se ha hecho una detección y se calcularán sus contornos.	26
Figura 8. Blobs a enviar al clasificador después de haberlos recortado de un frame en el que se han detectado como candidatos	26
Figura 9. Muestra de las imágenes utilizadas en las base de datos de entrenamiento del clasificador.	27
Figura 10. Esquema de bloques de la red neuronal Alexnet	28
Figura 11. Esquema de la salida de una capa para Resnet.	28
Figuras 12 y 13. Curvas de coste por época (izquierda) y curva de precisión de la clasificación de todas las imágenes respecto al total -	30
Figuras 14 y 15. Curvas de coste por época (izquierda) y curva de precisión de la clasificación de todas las imágenes respecto al total-	30
Figuras 16 y 17. Curvas de coste por época (izquierda) y curva de precisión de la clasificación de todas las imágenes respecto al total -	31
Figuras 18 y 19. Curvas de coste por época (izquierda) y curva de precisión de la clasificación de todas las imágenes respecto al total.	31
Figura 20. Ejemplo de la salida del bloque de clasificación por segundo. Promedia y calcula si hay mayoría de humo o no humo.	33
Figura 21. Ejemplo de la salida del bloque final del algoritmo. Indica para los últimos 10 segundos si ha habido humo o no humo	33
Figuras 22 y 23. Múltiples detecciones al moverse la imagen (izquierda). Detección de humo correcta en cami-cota2 (der)	36
Figura 24. Humo bien clasificado en Cantera	37
Figura 25. Blob de Humo detectado como no humo y dos blobs detectados correctamente como humo	38
Figura 26. Dos blobs de Humo detectados como humo(1) y un blob de humo detectado correctamente como humo(2)	38
Figura 27. Blob de Humo y no humo clasificados erróneamente para un test anterior con menos imágenes en base de datos	38
Figura 28. Blobs de humo y no humo detectados correctamente	38
Figura 29. Errores de clasificación de humo como no humo. Clasificación errónea (1) y clasificación errónea debido a que el recuadro en el que aparece el humo es muy grande y aparecen más elementos que no son humo.	39
Figura 30. Blobs de no Humo detectado correctamente (1) Blobs de humo y no humo correctamente clasificados	39
Figura 31. Falsas alarmas al clasificar elementos como humo siendo no humo. Frames aislados entre frames no humo	40
Figura 32. Blobs de no humo clasificados correctamente	40
Figura 33. Blob de humo clasificado como no humo	41
Figura 34. Clasificación del helicóptero entero como no humo	41
Figura 35. Humo generado por el avión clasificado correctamente como humo. Hélices del avión en movimiento humo erróneamente.	42
Figura 36. Blobs de no humo clasificados correctamente	42
Figura 37. Algunos blobs clasificados correctamente como no humo y falsa alarma generada por la puerta.	43
Figura 38. Humano clasificado erróneamente como humo	43
Figura 39. Blobs de objeto en movimiento correctamente clasificado como no humo	44
Figura 40. Blobs de persona clasificado incorrectamente. Vídeo sin estar en el contexto de paisajes.	45

Figura 41. Blobs de no humo detectados correctamente	45
Figura 42. Blobs detectados totalmente después de un cambio de imagen en la cámara	46
Figura 43. Blob de agua rompiendo en la orilla clasificado incorrectamente como humo(1) Blobs de nubes clasificados incorrectamente como humo	46
Figura 44. Blobs aislados detectados como no humo	47
Figura 45. Blobs clasificados correctamente como no humo	48

Lista de tablas:

En la lista de a continuación están definidas todas las listas que aparecen en el proyecto:

Tabla 1. Bloques de trabajo	13
Tabla 2. Resumen hitos	13
Tabla 3. Listado de vídeos	20
Tabla 4. Matriz de confusión del vídeo Camicota2.	36
Tabla 5. Matriz de confusión del vídeo Cantera	37
Tabla 6. Matriz de confusión del vídeo Carena	37
Tabla 7. Matriz de confusión del vídeo Carenataca	38
Tabla 8. Matriz de confusión del vídeo Llac2.	39
Tabla 9. Matriz de confusión del vídeo Pont	40
Tabla 10. Matriz de confusión del vídeo Playa	41
Tabla 11. Matriz de confusión del vídeo Cam-car.	42
Tabla 12. Matriz de confusión del vídeo Camino	43
Tabla 13. Matriz de confusión del vídeo Contorno.	44
Tabla 14. Matriz de confusión del vídeo Fondo Casa	44
Tabla 15. Matriz de confusión del vídeo Paisaje	45
Tabla 16. Matriz de confusión del vídeo Paisajes	46
Tabla 17. Matriz de confusión del vídeo VerdeCol.	47
Tabla 18. Matriz de confusión del total de vídeos con humo	48
Tabla 19. Matriz de confusión del total de vídeo sin humo	48
Tabla 20. Matriz de confusión del total de vídeos del proyecto	48
Tabla 21. Tabla de resultados finales de clasificación	49
Tabla 22. Tabla de resultados finales del analizador de vídeos	50

1. Introducción

En este apartado se definen los objetivos principales que se han definido para este proyecto, los resultados que se busca obtener para cumplir con los requerimientos, la base de la que parte el proyecto y el plan que se ha seguido durante todo el desarrollo del proyecto.

1.1. Objetivos del proyecto

Los objetivos principales en los que se ha basado este proyecto han sido los siguientes:

1.1.1 - Analizar las técnicas que se utilizaron en el proyecto de 2011 para la clasificación y detección del humo y compararlas con las técnicas actuales. Y a partir de este punto decidir las mejores opciones de implementación para este proyecto

1.1.2 - Implementar los bloques necesarios para la detección de humo utilizando las técnicas estudiadas del punto anterior. A raíz de esto se ha decidido que la parte de implementación del proyecto constará de dos grandes bloques:

- Bloque de extracción del fondo de la imagen:

En este bloque se analiza imagen a imagen cada video y se modela el fondo de la imagen para detectar objetos que se están moviendo. Los objetos detectados se procesan en el siguiente bloque, que se encargará de hacer una clasificación binaria de cada blob detectado en cada imagen.

- Bloque de clasificación de la imagen:

En este bloque se utilizan redes neuronales preentrenadas para hacer la clasificación de las imágenes y obtener un resultado de cada imagen.

1.1.3 - Obtener resultados a partir del testeo de videos y hacer un análisis de la calidad del algoritmo, del porcentaje de acierto en la detección del humo, de sus falsas alarmas y del conjunto global de los resultados

1.1.4 - Definir y proponer posibles mejoras a los algoritmos, técnicas y bases de datos utilizadas.

1.2. Requerimientos y especificaciones

El requerimiento principal de este proyecto consiste en que se detecte humo siempre que aparezca en los videos lo más rápido posible desde que se origina, que el número de falsas alarmas que se generen sea el mínimo posible, siendo la situación ideal de 0 falsas alarmas por video y que ha de ser un algoritmo que pueda procesar un número adecuado de imágenes por segundo.

Inicialmente se definió que se buscaría hacer una clasificación a tiempo real del humo, pero por limitaciones tecnológicas y por la complejidad de cálculo que tiene la red neuronal que se ha definido en el algoritmo, esto resulta más complicado con lo que se ha enfocado a que la clasificación del humo sea lo más acertada posible.

1.3. Procedimiento seguido:

Este proyecto está basado en continuar el proyecto que se inició el año 2011 de la UPC con AD Telecom, para encontrar un mecanismo que permitiese analizar videos y detectase humo y fuego.

Inicialmente se diseñó un bloque de detección de fuego, pero la necesidad de detectar el inicio de un incendio tiene mucho más impacto que cuando se está desarrollando, con lo que solamente se continuará con la parte de la detección de humo.

El procedimiento seguido partiendo de este punto ha sido el siguiente:

- Estudio de las técnicas actuales para la clasificación de humo, entre ellas la extracción del fondo de la imagen y el funcionamiento de las redes neuronales para la clasificación de imágenes.

- Desarrollo del algoritmo utilizando librerías de programación de visión computacional para análisis de imágenes y videos, Open CV sobre Python, y por otro lado la librería de diseño y creación de redes neuronales Pytorch.

Dentro del desarrollo se ha dividido el trabajo por bloques y para cada uno de ellos se ha definido una estructura entrada/salida con el resto de bloques del sistema. Cada bloque depende del resto de bloques

- Finalmente la extracción y análisis de resultados para la validación del sistema, analizando un conjunto de imágenes por video para cuantificar cada clasificación que se ha hecho y obtener unos resultados generales.

1.4. Plan de trabajo

El plan de trabajo se ha definido en los siguientes bloques de trabajo:

Instalaciones y búsqueda de información	WP ref: (WP#) 1
Mayor constituyente: software y búsqueda de información	1 de 4
En este primer bloque se hará la instalación de software necesario, librería de opencv y máquina virtual de Ubuntu. Se buscará información sobre el estado del arte en detección de humo.	Fecha inicio: 15/02/2018 Fecha fin: 14/03/2018
<ul style="list-style-type: none"> • Instalación OpenCV, python. • Búsqueda de información relacionada con detección de humos. Estado del arte • Estudio del algoritmo que se usó en 2011. 	

Diseño y creación del algoritmo	WP ref: (WP#) 2
Mayor constituyente: software	2 de 4
En este bloque se decidirán los bloques del algoritmo. Se trabajará en el diseño de los bloques y se harán las primeras pruebas con video.	Fecha inicio: 14/03/2018 Fecha fin : 21/04/2018

Bloque redes neuronales	WP ref: (WP#) 3
Mayor constituyente: software y búsqueda de información	3 de 4
En este bloque se aprenderá cómo funcionan las redes neuronales tanto en modo informativo como con el paquete pytorch de programación. Se harán pruebas con redes entrenadas para encontrar algunas que se adecuen a nuestra problemática	Fecha inicio: 22/04/2018 Fecha fin : 05/05/2018

Últimos test y memoria final	WP ref: (WP#) 4
Mayor constituyente: software y redacción	4 de 4
En éste último bloque se acabarán de ajustar parámetros del algoritmo para hacer los últimos test. Realizar los test finales Finalizar la memoria.	Fecha inicio: 5/05/2018 Fecha fin: 30/06/2018

Tabla 1. Bloques de trabajo

Hitos

WP#	Título	Hito / entregable	Fecha
1	Información e instalaciones	-	14/03/2018
2	Diseño algoritmo	Primeros algoritmos	21/04/2018
3	Bloque redes neuronales	Algoritmo y red entrenada	04/05/2018
4	Últimos test y memoria	Memoria y algoritmo final	30/06/2018

Tabla 2. Resumen hitos

Finalmente, éste es el diagrama de Gantt a modo resumen de las tareas definidas en los anteriores bloques de trabajo:

	Febrero	Marzo	Abril	Mayo	Junio	Julio
Inicio del proyecto						
Instalaciones y búsqueda de información						
Diseño y creación del algoritmo						
Bloque redes neuronales						
Últimos test y memoria final						
Presentación trabajo						

Figura 1. Diagrama de Gantt

1.5 Desviaciones respecto al plan de trabajo

Al principio del proyecto se definió un esquema basado en el proyecto que se realizó en 2011. En ese proyecto se utilizaron algoritmos de procesamiento de imagen para extraer características como el color o el movimiento del humo entre otros.

Como propuesta de la tutora se decidió, aproximadamente un mes después de comenzar la investigación, optar por utilizar técnicas más novedosas para la clasificación del humo y se decidió finalmente añadir un módulo de clasificación basado en redes neuronales. Eso conllevó al aprendizaje y búsqueda de información sobre su funcionamiento y por eso se comenzó un poco más tarde de lo inicialmente definido la implementación del bloque de clasificación.

Inicialmente la empresa ADTelecom nos propuso enviarnos videos con pruebas de humo pero al final no nos los pudieron proporcionar, con lo que para evaluar el algoritmo diseñado nos hemos basado en los videos que ya disponíamos del proyecto anterior i de otros encontrados en internet.

Por otro lado el plan de trabajo también ha tenido otra pequeña desviación con los test finales del algoritmo completo, ya que se han retrasado hasta las últimas semanas para poder encontrar el mejor modelo de clasificación de humo.

Finalmente se ha añadido un pequeño bloque que no estaba considerado en el inicio del proyecto para analizar los resultados de la clasificación de cara a la aplicación del algoritmo en la vida real.

2. Estado del arte de la tecnología aplicada en este trabajo

A día de hoy el procesamiento de imagen y video está en un momento de cambios continuos y existen muchas técnicas y formas de procesamiento.

En el caso específico de la detección de humos, la mayoría de sistemas están basados en el análisis de las características dinámicas que tiene el humo.

El humo como tal es un fenómeno físico y químico que cuando las sustancias se queman, se producen repentinamente cambios químicos y se generan partículas y gases por transformación o descomposición de las sustancias.

Estas partículas y gases son los que constituyen el humo.

El humo puede tener diferentes colores, debido a las diferentes composiciones que puede tener.

Debido al impacto del aire, el humo tiene características irregulares y difusas. Irregular, en el sentido de que su forma es siempre cambiante y difusa, ya que su área crecerá cada vez más. A medida que el área aumenta, la densidad disminuirá más y más.

El color del humo suele ser blanco-gris-negro, sin embargo, a medida que aumenta el área, el color del humo se vuelve más claro, ya que la densidad disminuye.

El gran conjunto de trabajos relacionados con la detección del humo se basan en el análisis de las características descritas y de la detección de movimiento en las imágenes, y esto se realiza modelando el fondo y comparando cada imagen con el modelo de fondo, dando pie a los dos bloques principales que suelen tener estos sistemas [10][13]:

1^{er} Bloque: Sustracción del fondo de la imagen más conocido como “background subtraction” en inglés y técnicas de detección de movimiento.

2nd Bloque: Detección y clasificación del humo que incluye métodos basados en la textura, color, forma, movimiento, energía y frecuencia entre otros. Y también el uso de redes neuronales para clasificación directa de las imágenes.

Cabe remarcar que las técnicas que se explicarán a continuación son válidas para cámaras de espectro visible, que son las que contempla este proyecto.

Existen otro tipo de algoritmos basados en cámaras de infrarrojos o imagen por satélite que no se tendrán en cuenta ya que no pueden ser aplicados por las características que tienen las imágenes que graban esas cámaras.

En anteriores trabajos de detección y clasificación de humo se han utilizado diferentes técnicas que se repiten de unos proyectos a otros y de las que se han ido buscando mejoras continuamente [12]. Las técnicas de detección del foreground utilizadas en estos trabajos [14] están basadas en su gran mayoría en el algoritmo de detección que parten del original de Stauffer i Grimson [3].

Para las técnicas de clasificación del humo finalmente se han hecho multitud de análisis sobre el color y la energía del humo [6] [12] extrayendo características del color del humo y por otro lado también se ha trabajado en el análisis de los diferentes espacios de color con elementos como el fuego y el humo [14]

2.1. Técnicas de detección de fondo y detección de movimiento

Existen varias técnicas utilizadas para detectar candidatos a humo: la detección del foreground y la detección de movimiento en la imagen.

Las técnicas de detección de foreground a comentar se basan en el estudio de las funciones de densidad de probabilidad de los píxeles y su comparación con los valores para los siguientes frames.

Por otro lado los sistemas de detección de movimiento analizan la diferencia entre una imagen y otra y si ha habido algún cambio en algún píxel (teniendo en cuenta un umbral) estos se considerarían candidatos a humo para nuestro caso

A continuación se comentan algunas de ellas y por último el algoritmo de Stauffer & Grimson que ha sido el utilizado en este proyecto para la detección de fondo

2.1.1. Detección de regiones en movimiento:

La detección de regiones móviles consiste en la extracción de regiones candidatas de humo al estimar las diferencias entre una región de una imagen con una imagen de fondo de referencia que se ha definido.

Un píxel se considerará en movimiento si la diferencia entre la intensidad del píxel analizado con el valor de ese mismo píxel en la imagen anterior supera un umbral.

Un píxel se consideraría en movimiento si:

$$|I(x,y,t) - I(x,y,t-1)| > T(x,y,t)$$

Donde $I(x,y,t-1)$ es la intensidad del píxel (x,y) en el $(t-1)$ th frame de I , y $T(x,y,t)$ es un valor de un umbral que se define con el experimento.

2.1.2. Estimación de fondo.

En éste método una imagen de fondo B_{n+1} en un instante de tiempo $n+1$ se estima recursivamente a partir de una imagen I_n y el fondo de la imagen B_n del video:

Se calcula de la siguiente forma:

$$B_{n+1}(x,y) = \begin{cases} aB_n(x,y) + (1-a)I_n(x,y) & (x,y) \text{ stationary} \\ B_n(x,y) & (x,y) \text{ moving} \end{cases}$$

Dónde $I_n(x,y)$ representa un píxel en el frame numero n del video y a es un parámetro entre 0 y 1. Los píxeles que se mueven se determinan extrayendo la imagen actual de la imagen de fondo.

$$X(x,y) = \begin{cases} 1 & \text{if } |I_n(x,y) - B_n(x,y)| > T \\ 0 & \text{otherwise} \end{cases}$$

T es un umbral que se define de acuerdo al estudio de la escena del fondo.

En el algoritmo definido, el siguiente paso consiste en un análisis de color para los píxeles que se han clasificado como foreground para decidir si un píxel es humo o no.

2.1.3. Stauffer & Grimson

Éste método es un estimador que permite utilizar hasta un número N de funciones Gaussianas por cada canal de color para estimar la función densidad de probabilidad del modelo de cada píxel y de esta manera crear un modelo de fondo [1] [2] [3].

Cada gaussiana está definida por su media y su varianza σ^2

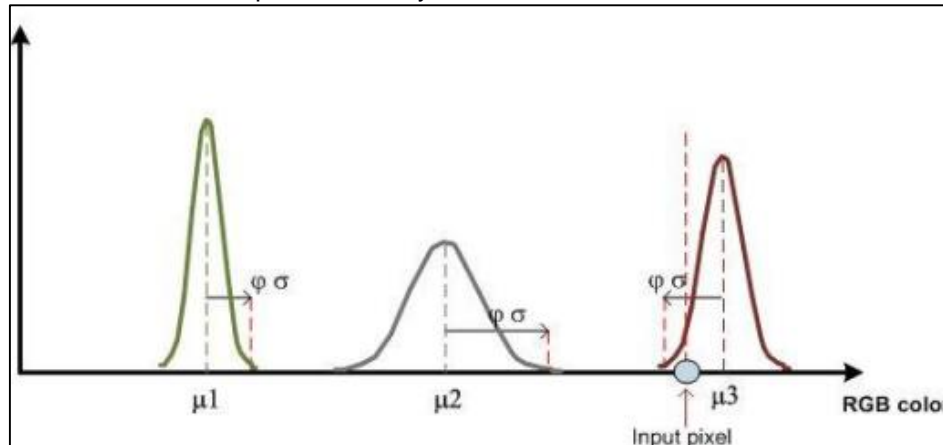


Figura 2. Ejemplo de tres gaussianas

La distribución de cada píxel en una imagen viene definida por la expresión:

$$P(I_t) = \sum_{k=1}^n \omega_{i,t} \cdot \eta(I_t, \mu_{i,t}, \sum i, t)$$

Donde n es el número de gaussianas utilizadas para modelar cada píxel (normalmente entre 3 y 5), $\omega_{i,t}$ es el peso asociado a cada gaussiana y el resto corresponde a la definición de la gaussiana de media μ_i

Para cada píxel de la imagen se genera un modelo de probabilidad del píxel modelado como una combinación de varias Gaussianas que están combinadas entre sí con un factor de ponderación $w_{i,t}$. Este factor de ponderación depende del número de veces que se da un valor en las gaussianas.

Finalmente, para cada píxel se tienen N Gaussianas cada una con sus pesos, media y varianza.

Se utiliza ésta estimación para obtener un modelo de píxeles de fondo y luego decidir si un píxel se corresponde con el modelo y por tanto si se trata de un píxel de fondo o por el contrario es primer plano.

Se ordenan las gaussianas de cada píxel de grande a pequeña dependiendo del ratio entre el peso w_i y la varianza. Con lo que la prioridad se calcula como:

$$p_i = w_i / \sigma_i$$

El modelo de fondo para cada píxel se modelará con las B gaussianas que tengan un mayor peso asociado y con menos varianza, es decir que p_i sean los mayores posibles:

$$B = \operatorname{argmin}_b \left(\sum_{k=1}^b \omega_{i,t} > T \right)$$

T es un umbral de decisión (normalmente 0.6) y B el número mínimo de gaussianas para cada píxel. Las gaussianas con menor prioridad se considera que modelan el color de los objetos de primer plano observados en el píxel.

El peso en una gaussiana representa las veces que aparece, para cada píxel, ese estado. Cuanto más aparezca un estado, la gaussiana irá aumentando el peso. Si un avión aterriza en un aeropuerto y se queda en la pista, el color no tendrá demasiado peso en el conjunto de gaussianas

ya que no aparece mucho. Pero cuando se avanza en el tiempo, el peso de este estado en la distribución del píxel aumentará ya que el avión ha quedado estático y aparece siempre.

Para tomar una decisión de si un píxel a clasificar corresponde al fondo de la imagen se toma en cuenta lo siguiente:

- Un píxel pasará a ser fondo si su valor corresponde a alguna de las B gaussianas ya que se considera que ya ha aparecido lo suficiente anteriormente como para considerarse de fondo.
- Un píxel pasará a ser primer plano y por tanto candidato a ser humo en nuestro algoritmo si su valor no corresponde a ninguna de las B gaussianas

Finalmente para cada píxel se actualizan la media y varianza y sus fórmulas son las siguientes:

$$\mu_t = \rho x_t + (1 - \rho) \mu_{t-1}$$

$$\sigma_t^2 = \rho (x_t - \mu_t)^2 + (1 - \rho) \sigma_{t-1}^2$$

Y para cada peso de las gaussianas se actualiza de la siguiente forma:

$$\omega_{k,t} = (1 - \alpha) \omega_{k,t-1} + \alpha (M_{k,t})$$

Donde $M_{k,t}$ es 0 para cuando la gaussiana no ha coincidido con el modelo y 1 para cuando sí que ha coincidido.

Si un píxel no ha encajado con ninguna de las gaussianas, se crea una gaussiana en el valor del color de ese píxel con un peso muy bajo. Y si éste valor vuelve a repetirse, el peso de esta gaussiana se va actualizando hasta que el peso de la gaussiana es igual o más grande que el peso del resto de gaussianas. Para el próximo frame que el píxel vuelva a tener un valor parecido al de la nueva gaussiana, ya dejará de considerarse primer plano y pasará a ser del modelo de fondo.

Con este algoritmo se permite detectar todos aquellos objetos que no pertenecen al fondo de la imagen y que por tanto serán los candidatos a ser analizados de si son o no son humo.

Finalmente se ha decidido utilizar éste algoritmo ya que los resultados de éste modelo están aplicados en multitud de proyectos por el mundo con resultados muy buenos y porque está totalmente optimizado para la librería de Computer Vision (OpenCV) utilizada para la creación del algoritmo de detección de humo.

2.2. Clasificación de las imágenes

Para la clasificación de las imágenes se aplican dos metodologías actualmente. Unas basadas en el análisis sobre las características del humo para las que aplicando ciertos umbrales se decide si hay humo o no en una imagen [9] [11], y por otra parte se utilizan redes neuronales para que una imagen de entrada (en este caso un frame o bien un recorte de este frame) se clasifique con la red entrenada. A continuación se explican algunas de estas técnicas

Técnicas de análisis de características del humo y su color:

2.2.1. Análisis sobre el color del humo:

Cada píxel de la imagen se analiza y se obtiene su valor RGB. Este valor puede ser convertido en luminancia y crominancia, en el espacio de color YUV o bien en HSV. Cada color tiene un rango de valores entre [0 - 255]

La mayoría de las muestras de humo tienen un color grisáceo. Así que se pueden formular los píxeles de humo de la siguiente manera, donde Th es un umbral global que va de 15 a 25. Donde las componentes de color para ese umbral son muy parecidos al color que toma el humo.

$$|R(x, y) - G(x, y)| \leq Th$$

$$|G(x, y) - B(x, y)| \leq Th$$

$$|R(x, y) - B(x, y)| \leq Th$$

La ecuación establece que los píxeles de humo deben tener similares intensidades en sus canales de color RGB, con lo que los píxeles detectados en la imagen que cumplan estas características serían clasificados como humo. Hay que tener en cuenta que antes de aplicar ésta fórmula se deben limitar los candidatos a humo aplicando alguna de las técnicas de extracción de fondo u otros.

2.2.2. Análisis de componentes conexas

El humo suele aparecer en bloques, con lo que se busca encontrar objetos en la imagen que estén conexos entre ellos. Por la naturaleza de las imágenes, los píxeles que cumplan patrones de color y movimiento tenderán a agruparse en regiones de píxeles conexos, cada una de estas regiones corresponderá a elementos físicos. Estos elementos físicos por tanto serían candidatos a ser humo.

Aplicación de las redes neuronales a la clasificación de humo:

Para varios proyectos se han utilizados redes neuronales [4] [8] a la hora de clasificar candidatos a humo. Ésta clasificación consiste en utilizar una red neuronal que se entrene con una base de datos con multitud de imágenes con humo y otras sin humo [7].

El proceso es similar a cualquier tipo de clasificación con redes neuronales, se utilizan imágenes de una base de datos que se iteran durante un número limitado de épocas buscando encontrar una minimización de la función de coste y conseguir la máxima precisión posible en la clasificación.

Se utilizan estas redes para analizar imágenes tanto de bosques con cámaras digitales y fijas como para imágenes recogidas por satélite.

En el apartado 3.3 se explican las redes neuronales utilizadas en este proyecto con más detalle.

Las técnicas que se definirán son las redes Alexnet, que son unas redes convolucionales, y las redes Resnet, que son unas redes neuronales profundas que trabajan con el residuo que se obtiene de una capa de la red a la siguiente.

Se ha decidido utilizar las redes neuronales como método de clasificación para el humo ya que son técnicas muy modernas y que las técnicas de análisis de características del humo se llevan utilizando muchos años y ya se han hecho muchos estudios con ellas.

3. Metodología/Desarrollo del proyecto:

El algoritmo ha sido desarrollado en el lenguaje de programación python, con el uso de la librería de Computer Vision OpenCV y para el bloque de clasificación de las imágenes la librería Pytorch.

A continuación se describen brevemente sus funciones:

- Lenguaje de programación Python: Se trata de un lenguaje que permite crear programas usando varios estilos de programación, como la programación orientada a objetos y la programación imperativa, y que se adapta a las características del proyecto ya que permite cargar el módulo de visión artificial para el análisis de imágenes y videos.

- Librería de Computer Vision OpenCV: Es una librería libre que pretende proporcionar un entorno de desarrollo fácil de utilizar y muy eficiente y que permite multitud de operaciones como la detección de movimiento y la extracción de fondo de las imágenes, punto clave en el proceso de detección de humo.

- Pytorch: Es una librería libre de “machine learning” que permite al usuario crear sus propias redes neuronales o utilizar redes ya existentes para su uso en clasificación.

Utilizando estas tres herramientas se ha conseguido integrar en un solo algoritmo todo el sistema de detección y clasificación que se buscaba.

Se han utilizado dos bases de datos para este proyecto, una base de datos de imágenes buscadas en internet para el entrenamiento y validación de la red neuronal y una base de datos de vídeos que consta de varias escenas forestales:

A. Base de datos para el entrenamiento de la red neuronal:

Está compuesta por dos particiones de datos: entrenamiento y validación. Entrenamiento contiene 586 imágenes clasificadas como humo y 1059 como no humo, y para validación el número es bastante menor para ambos casos y tiene 54 imágenes como humo y 48 como no humo. Cada una de estas imágenes no es una imagen entera sino que son partes pequeñas de otras imágenes y se han obtenido de internet (Google imágenes entre otros).

El total de imágenes enteras utilizado ha sido de 110.

El número total con el que se ha entrenado la red es de 1741 imágenes.

Más adelante, en el apartado 3.4 se entrará más en detalle de las imágenes utilizadas.

B. Base de datos de vídeos:

Está compuesta por 13 secuencias de vídeo, 6 de ellas en las que existe humo y 7 en las que no. Las secuencias con humo son en su mayoría escenas forestales en las que dado cierto segundo comienza a aparecer una columna de humo y en las de no humo son secuencias obtenidas de internet en las que existe movimiento en algunos paisajes.

Se han utilizado dos tipos diferentes de vídeos, para poder analizar si la clasificación y detección del humo era correcta en las tramas con humo y para detectar posibles falsas alarmas para un caso general de análisis de videos:

Número de sec.	Nombre video	Humo/No humo
1	CamiCota-2.avi	Humo
2	Carena.avi	Humo

3	CarenaTaca.avi	Humo
4	Pont.avi	Humo
5	Cantera.avi	Humo
6	Llac-2.avi	Humo
7	Cam-car.avi	No humo
8	Camino.avi	No humo
9	Contorno.avi	No humo
10	Fondo-casa.avi	No humo
11	Paisaje.avi	No humo
12	Paisajes.avi	No humo
13	Playa.avi	Humo
14	Verde-col.avi	No humo

Tabla 3. Listado de vídeos

De cada uno de los videos se han escogido los intervalos de tiempo iniciales para hacer el análisis ya que en los que aparece el humo, el tiempo en el que tarda en aparecer es bastante corto (aproximadamente menos de 15 segundos). El tiempo de análisis que se ha escogido para cada vídeo es de aproximadamente 1 minuto. Esto varia con cada vídeo ya que en algunos se hacen múltiples detecciones y en otros muy pocas, con lo que se ha decidido el período a analizar en función de la salida que se ha ido extrayendo.

3.1. Esquema general:

Para volver al contexto inicial una vez definida la base de datos con la que se ha trabajado, el objetivo consiste en, a partir de proporcionar un video en la entrada, se indica si hay humo o no en las imágenes clasificadas. Para ello se han diseñado varios bloques en los que se divide el algoritmo.

Diagrama de bloques del algoritmo:

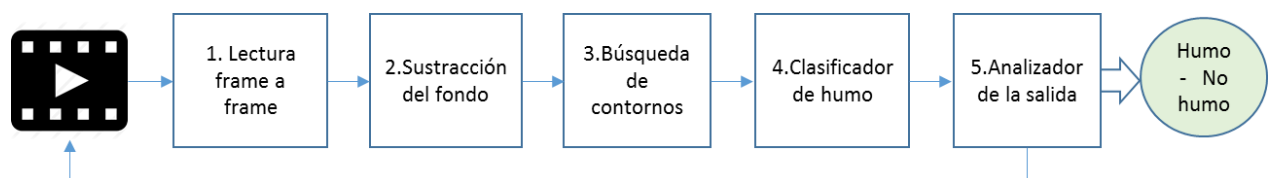


Figura 3. Esquema general del algoritmo

1. *Lectura “frame” a “frame” del video*: El algoritmo comienza haciendo una lectura del video que se quiere analizar y se introduce en un bucle que permite leer imagen a imagen el video.

2. *Sustracción del fondo de la imagen*: Este segundo bloque es uno de los más importantes del algoritmo ya que es de donde se extraerán los posibles candidatos a humo.

Para cada secuencia de videos tiene la función de detectar elementos nuevos respecto a los frames anteriores del video, que se convertirán en posibles humos.

3. *Búsqueda de contornos en la máscara binaria*: Del bloque anterior se genera una máscara binaria donde se ha detectado un elemento nuevo para cada frame, este bloque busca los contornos de cada elemento nuevo y extrae una imagen para cada elemento nuevo que se enviará al clasificador.

4. *Clasificador de humo*: este bloque analiza a que clase (humo / no humo) pertenecen los fragmentos de imagen detectado en el punto 2, para ello se ha utilizado un modelo de clasificación basado en redes neuronales, en este caso se ha utilizado la conocida red AlexNet preentrenada.

5. *Analizador de la salida del algoritmo*: Finalmente, este bloque sirve para hacer un análisis enfocado a tener una aplicación donde introducir este algoritmo, para hacer una decisión de si hay o no hay humo en un intervalo de tiempo.

A continuación se explicará todo el detalle de cada uno de los puntos del 2 al 5 del diagrama de bloques así como las pruebas que se han hecho de cada uno de los apartados. El punto uno es únicamente un bucle para recorrer todo el video imagen a imagen y un redimensionado de cada imagen para analizar todos los videos por igual.

3.2. Sustracción del fondo:

La sustracción de fondo es una técnica que permite detectar nuevos objetos en un video, asignando a cada píxel si forma parte del fondo de la imagen (background) o si está en un primer plano (foreground).

Para cada frame inicialmente se busca encontrar todos los elementos que se están moviendo. Con lo que todos los elementos que sean nuevos se considerarán primer plano y serán los posibles candidatos a ser humo y el resto se considerarán fondo, que aplicado a este caso serían los bosques/edificios/construcciones que aparecen en los videos.

La definición de este modelo de fondo tradicionalmente requeriría de un tiempo de observación para que se definiese el modelo de fondo pero para este caso el objetivo de este bloque es detectar cualquier elemento que se esté moviendo, con lo se empieza a calcular el modelo fondo de cada uno de los píxeles desde el momento que se inicia el video sin tener que hacer ningún entrenamiento inicial.

El modelo parte de los valores individuales de cada píxel, con lo que depende exclusivamente de cada píxel por separado y no tiene en cuenta los elementos que tenga cada píxel a su alrededor

El algoritmo utilizado para la sustracción del fondo en este algoritmo se basa en una función proporcionada por la librería de computer visión OpenCV.

3.2.1. Modelo de imagen con varias gaussianas (MOG2)

El método de sustracción de fondo que se utiliza en este proyecto está basado en dos papers de Z.Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction" en 2004 [1] y "Efficient Adaptive Density Estimation per Image Pixel for the Task of Background Subtraction" [2] en 2006 y ambos están basados en el algoritmo de Stauffer y Grimson (Stauffer y Grimson 1999) [3], en el que se usan varias Gaussianas por píxel.

En el momento en que se comienza a analizar el vídeo, se crea un modelo de fondo para cada píxel del primer frame. Esto quiere decir que se asigna una función densidad de probabilidad (fdp) con varias Gaussianas para cada píxel de la imagen de cada uno de los canales de la imagen, en este caso tres (RGB).

El número de gaussianas utilizado por píxel ha sido de 3 para todos los vídeos.

El procedimiento de cómo se calcula éste modelo de fondo y su funcionamiento se ha explicado en el apartado 2.1.3.

A partir de este punto se considera que se han creado modelos de fondo con tres gaussianas para cada vídeo y que se ha trabajado con la máscara binaria obtenida.

3.2.2. Modelo de fondo utilizado

Se han calculado dos modelos de sustracción de fondo, uno en el que se obtienen todos y cada uno de los elementos nuevos del vídeo por imagen y otro para el que se busca encontrar los elementos de la imagen para los que el valor del píxel es muy diferente al del modelo de fondo.

El objetivo de este primer modelo de fondo es por tanto obtener todos los elementos nuevos, comportamiento habitual de un modelo de sustracción de fondo.

Y el objetivo de éste segundo modelo es detectar los elementos que son muy diferentes al humo.

Esto es así ya que estamos asumiendo que el humo es poco denso y cambiará poco el color de los píxeles de un frame al siguiente y también respecto al modelo de fondo.

El humo es un elemento que crece lentamente desde que se origina con lo que no se verá reflejado en este segundo modelado del fondo ya que la variación de color para un píxel en el que comienza a aparecer humo se hace progresivamente como ya se ha comentado.

OpenCV permite a la hora de calcular un modelo de fondo, establecer un umbral fijo para cada píxel en el que si se supera ese umbral, se muestre ese píxel en la máscara binaria de la detección de primer plano. El valor asociado a superar ese umbral se define como la distancia entre el valor actual de ese píxel con las gaussianas. Este procedimiento no afecta a la actualización del fondo.

La distancia que se utiliza para este caso es la raíz cuadrada de la distancia de mahalanobis entre el píxel y el modelo de fondo y ha de ser mayor al umbral definido en la siguiente fórmula:

$$d(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T S^{-1} (\vec{x} - \vec{y})}.$$

El umbral se ha calculado en base a hacer varias pruebas y observar los resultados para los casos más relevantes, por defecto es 16 (que no afecta al modelo) pero se ha establecido a 100 para éste segundo modelo. Por lo tanto, el umbral definido únicamente sirve, una vez calculado el modelo de fondo para un frame, para mostrar los píxeles que son primer plano y que están superando la distancia definida para el umbral, en otras palabras, sólo muestra aquellos píxeles en que la diferencia con el fondo es muy grande.

Finalmente se aplica una diferencia/resta entre la imagen binaria del primer modelo con el segundo.

Para así generar una máscara binaria en la que se buscarán los posibles candidatos a humo y se habrán eliminado varios elementos que resultan ser perjudiciales para nuestro sistema ya que no tienen ningún parecido, en cuanto a comportamiento en un vídeo, a la forma y densidad del humo.

· Máscara binaria final = Máscara binaria fondo del 1er modelo – Nueva imagen binaria calculada con el umbral definido sobre el modelo fondo del segundo modelo.

Resultado: Máscara en la que se han eliminado los elementos que han sido detectados como primer plano en el segundo fondo y que superaban el umbral definido. Ésta máscara es solo a nivel de análisis y búsqueda de candidatos, esto no afecta a la actualización del fondo de los modelos.

A continuación, a cada imagen binaria obtenida por la aplicación del sistema de detección de primer plano se le aplica una operación llamada apertura morfológica, que es una operación que se utilizará para remover los objetos pequeños del modelo de fondo de la imagen.

Esta función se compone de dos pasos, una erosión seguida de una dilatación aplicando un elemento estructurante B, donde un elemento estructurante es un bloque de píxeles binarios que tiene una forma concreta. En este caso se ha utilizado como elemento estructurante una elipse (5-5) que tiene la siguiente forma:

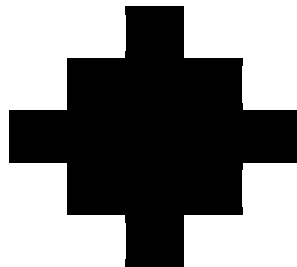


Figura 4. Elemento estructurante, elipse (5 – 5)

La apertura de A por B se obtiene por la erosión de A por B, seguida por la dilatación de la imagen resultante por B, donde A es la imagen y B es el elemento estructurante.

La erosión sigue la siguiente fórmula:

$$A \ominus B = \{x \mid x + b \in A \text{ for every } b \in B\}$$

La erosión se define como el proceso de tomar copias de A y trasladarlas por los vectores de movimiento definidos para cada píxel de B, en este caso moviéndolas en la dirección contraria (- b) y hacer la intersección entre ellas. Éste proceso se escribe cómo:

$$A \ominus B = \bigcap_{t \in B} A_{-t}$$

El resultado de ésta operación consistiría por tanto en comprobar si el elemento estructurante B está completamente contenido en A. En caso contrario el resultado sería el conjunto vacío y se pondría como color negro el píxel para el que se haya calculado el conjunto vacío al hacer la intersección.

Por otro lado la dilatación sigue la siguiente fórmula:

$$A \oplus B = \{c \mid c = a + b \text{ for some } a \in A \text{ and } b \in B\}$$

La dilatación se define como el proceso de tomar copias de A y trasladarlas en los vectores de movimiento definido para cada píxel de B (con el origen de B en este caso en el centro de la elipse). Al unir estas copias se obtiene la dilatación de A por B:

$$A \oplus B = \bigcup_{t \in B} A_t$$

El resultado de esta operación resultaría en la expansión de los píxeles de A con la forma del elemento estructurante B.

Se tiene que tener en cuenta que para estas operaciones solo se consideran los píxeles blancos de la máscara binaria y los elementos en negro del elemento estructurante pero aplicando la imagen complementaria para éste último, donde los píxeles que son negros del elemento estructurante serían blancos.

Finalmente si se aplica una función seguida de la otra obtenemos la fórmula de la apertura, que elimina los elementos que son más pequeños que el elemento estructurante con la erosión y luego con la dilatación reconstruye los elementos que eran mayores que el elemento estructurante para volver a la forma inicial.

Finalmente la fórmula de la apertura se define cómo: $A \circ B = (A \ominus B) \oplus B$

· Ejemplo de apertura con una imagen:



Figura 5. Ejemplo de apertura morfológica

Con lo qué, aplicando esta operación, eliminamos el posible ruido que pudiese aparecer en el modelo de fondo y pequeños puntos aislados.

Una vez definidas las técnicas aplicadas en este bloque, el resultado que se obtiene es de una máscara binaria para cada frame de la cual se han eliminado, con la apertura morfológica, los elementos aislados y muy pequeños y que por otro lado se han eliminado los elementos más rápidos, utilizando la diferencia con el modelo de fondo auxiliar.

Teniendo en cuenta estas consideraciones el algoritmo deberá enviar al bloque de clasificación los elementos restantes detectados, pero para ello se utiliza la detección de contornos que se define en el apartado siguiente.

A continuación se muestran unos ejemplos de máscaras binarias calculadas en este bloque:

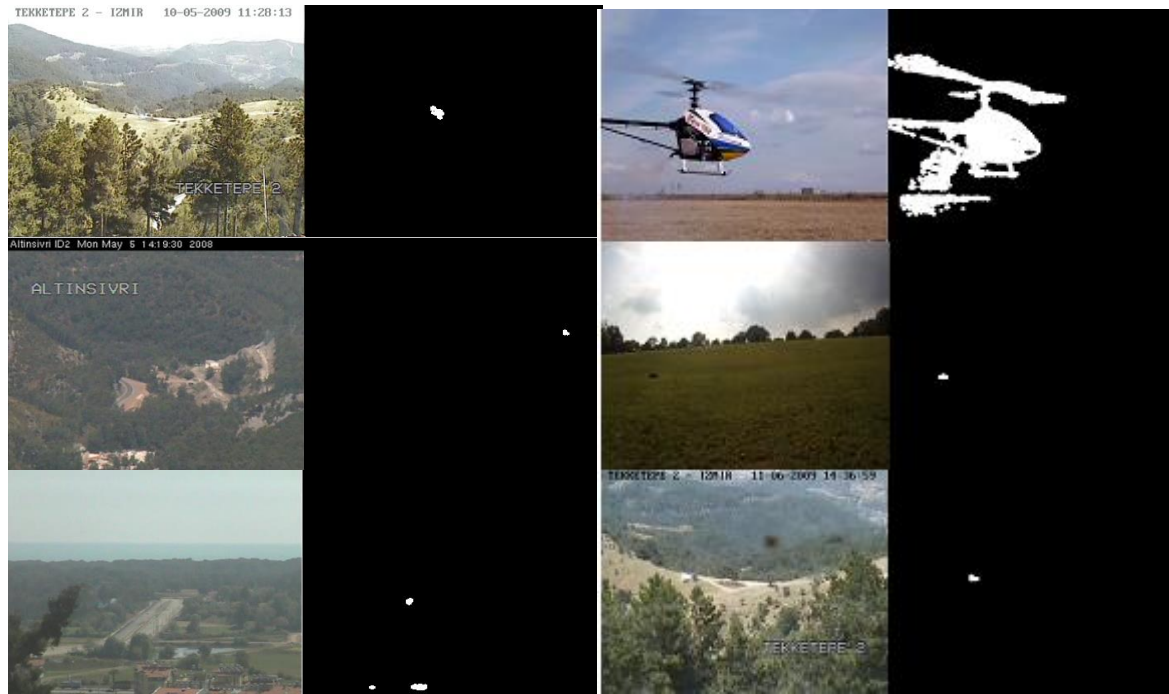


Figura 6. Ejemplo de sustracción de fondo para un frame de algunos vídeos utilizados

3.2.3. Conclusiones sobre el bloque de sustracción de fondo de la imagen

Con este bloque se consigue obtener los mejores candidatos a humo ya que se eliminan elementos rápidos y el ruido de la imagen, dejando sólo elementos que puedan cumplir características de movimiento y tamaño similares al humo.

El fondo de la imagen se va actualizando constantemente y no deja de aprender en ningún momento. Esto supone que para un vídeo en el que se lleva detectando correctamente humo durante varios segundos, las nuevas detecciones serán de zonas en las que aparece humo nuevo y se deja de detectar el humo que se había detectado inicialmente, es decir, se detecta el humo que ha continuado desplazándose por la escena. El hecho de que el modelo aprenda cada vez que hay un cambio es bueno ya que si aparecen nuevos elementos en el vídeo y no se mueven esto permitirá que cuando se haya adaptado el modelo, éste nuevo elemento no aparezca en la máscara binaria y por tanto que no se envíe al clasificador.

Un ejemplo sería el de un vehículo que se estaciona en un parking.

Por el contrario, no aprender permite detectar todo el humo en lugar de un fragmento del humo pero a la vez podría afectar los resultados de su clasificación al ser cada vez más grande la región detectada.

Esto podría parecer un problema pero el humo tiende a crecer, con lo cual mientras exista humo en la imagen, este se seguirá detectando hasta que se extinga.

Finalmente, teniendo en cuenta lo comentado en la última línea, el objetivo es detectar si existe humo en una imagen y no detectar todo el humo de una imagen que proviene de un mismo foco y esto se cumple en la mayoría de ocasiones.

3.3. Búsqueda de componentes conexas

Para este apartado partimos del punto anterior en el que para cada frame del vídeo se va obteniendo una máscara binaria, en esa máscara binaria pueden existir o no elementos detectados, con lo cual sólo se buscarán las componentes conexas para los elementos que se hayan detectado en la máscara binaria.



Figura 7. Ejemplo de máscara binaria en la que se ha hecho una detección y se calcularán sus contornos.

Se ha utilizado una función de OpenCV que permite encontrar para una imagen binaria los elementos conexas a partir de la búsqueda de sus contornos. La segmentación por componentes conexas consiste en agrupar como un solo elemento todos aquellos píxeles individuales de detección que estén conectados entre. En este caso si se observa el ejemplo anterior se buscaría encontrar el contorno de los elementos en blanco que están conexas y que pertenecen a nuevos elementos en el vídeo.

El procedimiento que se sigue para detectar componentes conexas es el siguiente:

Se recorre la imagen de izquierda a derecha y de arriba abajo y se hace un rastreo de píxeles. El primer píxel en blanco encontrado se etiqueta como blanco. Se continúa al siguiente píxel a la derecha y se observan los píxeles que lo envuelven (adyacentes para todas las direcciones, conectividad 8). Si el color de éste píxel es blanco y alguno de sus vecinos es de su mismo color, se etiqueta como blanco.

Se recorre toda la imagen dando etiquetas nuevas para cada píxel blanco aislado y se sigue con éste procedimiento

Finalmente se utilizan las coordenadas en las que aparece cada componente conexo de la imagen para encontrar las posiciones de x e y máximas y mínimas de cada elemento.

De esta forma se puede definir un rectángulo a su alrededor para posteriormente buscar ese recuadro en el frame original del video y extraer un recorte que será enviado finalmente al clasificador del siguiente bloque.

A continuación se muestra un ejemplo de imágenes recortadas que se han enviado al clasificador para alguno de los vídeos:



Figura 8. Blobs a enviar al clasificador después de haberlos recortado de un frame en el que se han detectado como candidatos

El número de fragmentos de imagen a enviar por frame se ha limitado a 5 ya que cuando se produce un cambio radical en la imagen, al actualizarse el fondo de la imagen se detectan muchos nuevos objetos como “foreground” que serían enviados a analizar. Limitando a cinco este campo permite al algoritmo continuar procesando los siguientes frames del video sin quedarse en un frame haciendo una gran cantidad de clasificaciones que carecerán de sentido ya que provienen de un cambio o movimiento brusco en la secuencia.

Estos cambios radicales de fondo pueden surgir cuando la cámara se enfoca a otros lugares o bien se produce un movimiento en la cámara que hace se agite.

3.4. Clasificador de humo

Para clasificar el humo se ha utilizado una red neuronal que permite, a través de una imagen de entrada cualquiera, decidir si esta imagen pertenece a la clase “Humo” o la clase “No humo”. Por tanto, se han definido dos clases en la salida del clasificador.

La librería de “machine learning” Pytorch es una herramienta extensa y utilizada en grandes proyectos que permite crear redes neuronales desde cero (entre muchas otras funcionalidades) y definir cada una de sus capas interiores. Para éste algoritmo se ha utilizado una red neuronal preentrenada llamada AlexNet, que consiste en una red en la que ya vienen inicializados los parámetros de cada una de las capas y nodos de la red y que es muy útil en casos que la base de datos que se utiliza está limitada, que es precisamente el caso de este proyecto.

Se busca entrenar de nuevo la red preentrenada para que se adapten sus parámetros a los del humo.

El procedimiento consiste en cargar los parámetros calculados en los nodos y capas de la nueva red neuronal e iniciar un ciclo de entrenamiento con la nueva base de datos. Éste procedimiento sería el equivalente a volver a hacer una iteración para todas las imágenes de una base de datos durante un número determinado de épocas, 25 para nuestro caso, para una red nueva, con la diferencia de que en este caso ya viene inicializada.

Para entrenar la red neuronal preentrenada se ha generado una base de datos con temática de humo.

Se ha utilizado una base de datos creada manualmente con 1741 imágenes en total que se han dividido entre entrenamiento y validación. Cada una de ellas ha sido clasificada por su clase, humo o no humo.

· Ejemplo imágenes utilizadas para el humo y el no humo en el entrenamiento:

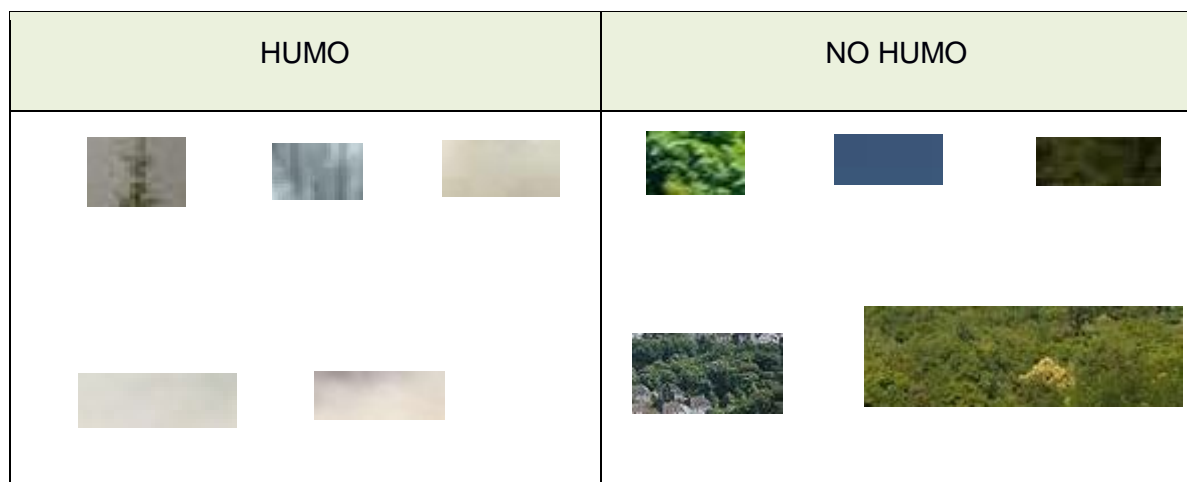


Figura 9. Muestra de las imágenes utilizadas en las base de datos de entrenamiento del clasificador.

Para el entrenamiento se han utilizado 586 imágenes etiquetadas como humo y 1059 como no humo. En la validación se han utilizado muchas menos imágenes, 54 etiquetadas como humo y 48 como no humo.

Las de entrenamiento son imágenes que se utilizan para calcular los pesos asignados a cada una de las capas y nodos de la red neuronal, que suelen ser la gran parte de las imágenes de la base de datos, y , por otro lado, las de validación, que han servido para comprobar y testear la clasificación con nuevas imágenes.

Estas imágenes se han obtenido de la búsqueda por internet de imágenes con paisajes, bosques y ciudades en su mayoría, que son lugares candidatos a ser monitorizados para un control de incendios y también de imágenes de incendios para el humo.

De figuras grandes se han hecho recortes más pequeños para tener una base de datos con imágenes relativamente pequeñas para parecerse a la entrada que se le envía al bloque de clasificación.

En las primeras pruebas del algoritmo se utilizaron 500 imágenes de no humo en el entrenamiento en vez de las 1059 que se han utilizado finalmente. Para este caso se detectaron muchas falsas alarmas en las que aparecían objetos que se clasificaban como humo y en realidad no lo eran. A raíz de esto se decidió añadir varias imágenes más al entrenamiento del “no humo” para ver si con más imágenes de esta casuística la clasificación de estos casos mejoraba, ya que la clase no humo abarca un mundo de imágenes mucho más grande que las imágenes que son exclusivamente humo.

Y finalmente con éste añadido se ha conseguido mejorar la clasificación para muchos de los casos donde se habían generado falsas alarmas.

Inicialmente se decidieron dos opciones a la hora de escoger qué tipo de red era el adecuado para la clasificación de imágenes. Se propusieron dos tipos de red con las que se han hecho varias pruebas hasta decidirse finalmente por una opción.

Estos dos tipos de redes son de los más utilizados en clasificación y son AlexNet y ResNet:

- AlexNet: es el nombre de una red neuronal convolucional, escrita originalmente con CUDA para funcionar con soporte de GPU, que compitió en el desafío de reconocimiento visual de gran escala ImageNet en 2012 y obtuvo grandes resultados. AlexNet contiene ocho capas; las primeras cinco son capas convolucionales, y las tres últimas son capas conectadas por completo.

El esquema de su red es el siguiente [2]:

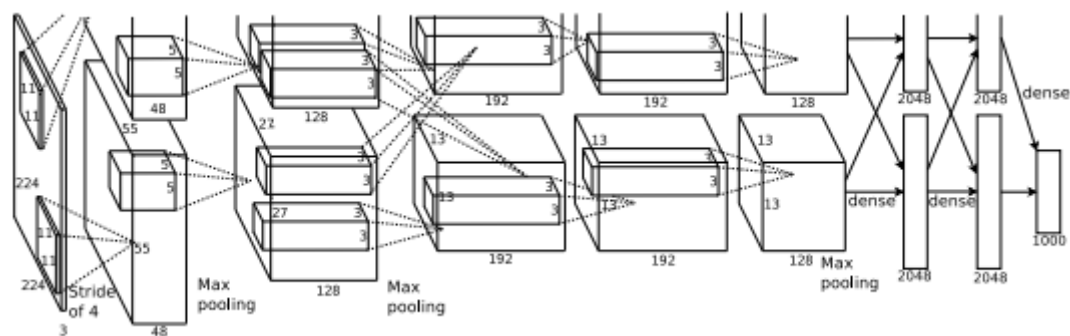


Figura 10. Esquema de bloques de la red neuronal Alexnet

Se introduce la imagen en la entrada y pasa a través de los cinco bloques convolucionales, para finalmente llegar a las capas conectadas y en nuestro caso en la última capa solo tener dos variables de salida y no las 1000 que se utilizaron en el 2012 en el concurso de ImageNet. En la salida había 1000 nodos ya que se habían definido 1000 clases para las diferentes imágenes de su base de datos.

- ResNet18: es el nombre de una red neuronal profunda residual. Este tipo de red se basa en que cada capa subsiguiente en la red neuronal profunda solo es responsable de ajustar

la salida de una capa previa simplemente agregando un "residuo" aprendido a la entrada. Esto difiere de un enfoque más tradicional en el que cada capa tenía que generar la salida deseada.

El esquema de un bloque de construcción del residuo es el siguiente:

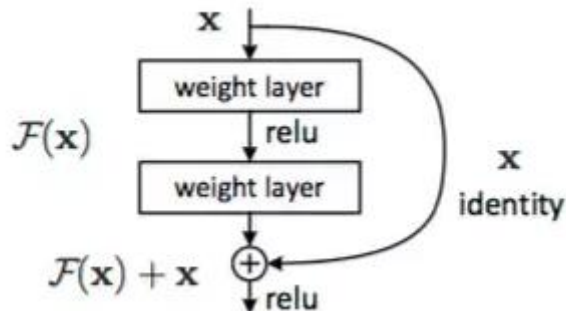


Figura 11. Esquema de la salida de una capa para Resnet.

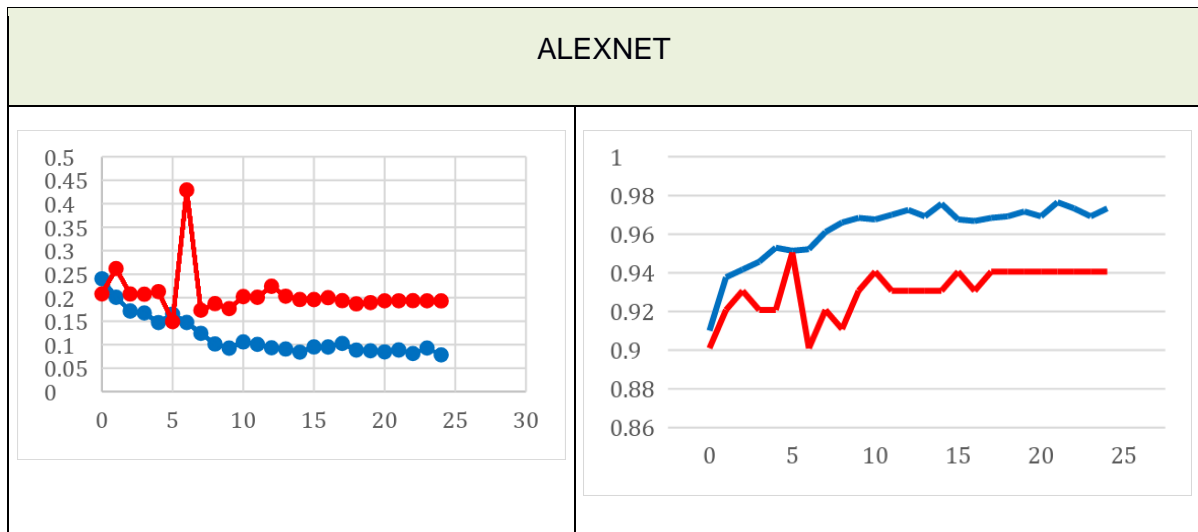
Las características de ésta red neuronal es que son más sencillas de optimizar y que se permiten adaptar a entrenamientos complicados.

Para ambos casos se han utilizado los modelos de estas redes preentrenadas y definidas para la librería Pytorch y utilizando dos variantes, que serían utilizando el ciclo completo de capas de las que se compone cada red y utilizando el método de congelación de capas, que permite no actualizar los parámetros de las capas congeladas dejando sólo la última capa como variable. Ésta última capa es totalmente conexa, esto implica que es la única en la que se aplica una operación lineal en el vector de entrada para decidir la clase a la que pertenecerá esa entrada.

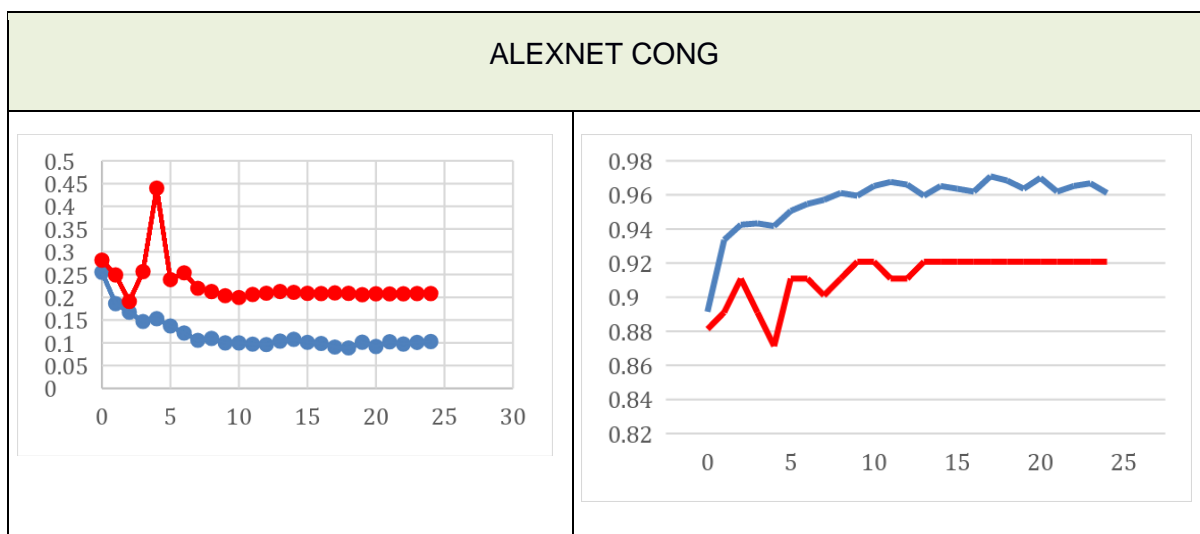
Se decidió comprobar si el método de congelación de capas daba mejores resultados en la precisión pero no ha sido así, y esto se ha comprobado revisando los resultados de la clasificación en la Figura 12.

Para decidir qué modelo de red y que tipo de función utilizar se han revisado los resultados de la precisión de la clasificación respecto al total de imágenes de la base de datos (% de acierto entre número de imágenes bien clasificadas entre el total de imágenes) para cada una de estas redes con las imágenes de entrenamiento y validación- Es importante detectar correctamente como humo las imágenes que realmente son humo pero también lo es lo contrario, detectar correctamente elementos que no son humo como no humo.

- Los valores en azul son los resultados para las imágenes de entrenamiento y los rojos para la validación

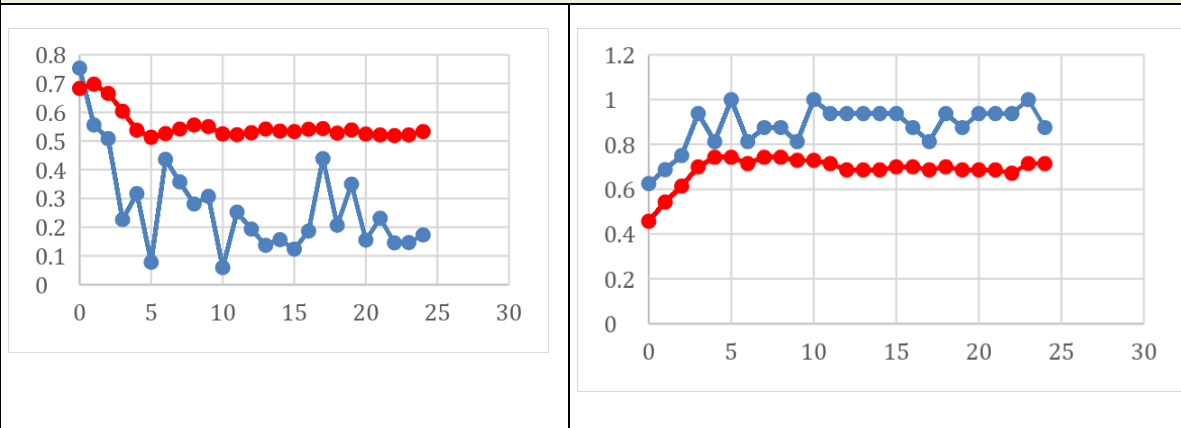


Figuras 12 y 13. Curvas del coste por época (izquierda) y curva de precisión de la clasificación de todas las imágenes respecto al total.



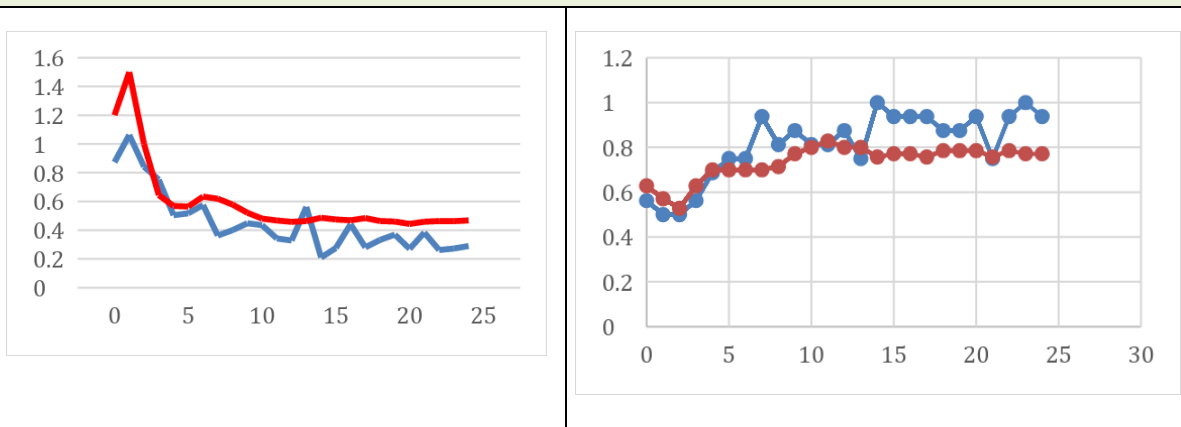
Figuras 14 y 15. Curvas del coste por época (izquierda) y curva de precisión de la clasificación de todas las imágenes respecto al total.

RESNET18



Figuras 16 y 17. Curvas del coste por época (izquierda) y curva de precisión de la clasificación de todas las imágenes respecto al total.

RESNET18 CONG



Figuras 18 y 19. Curvas del coste por época (izquierda) y curva de precisión de la clasificación de todas las imágenes respecto al total.

En las gráficas anteriores se han definido dos funciones, la precisión respecto al total de imágenes clasificadas (derecha) de la clasificación para el entrenamiento y la validación a lo largo de 25 épocas (número de veces que se itera el algoritmo para buscar los mejores resultados) y por otro lado la curva del coste (Loss) por época (izquierda). Cuando se habla del coste se refiere al mapeo que se hace con los valores obtenidos con un número real que represente el coste asociado a la clasificación hecha, por lo tanto se busca minimizar éste valor. La función de coste utilizada ha sido la pérdida de entropía cruzada, que mide entre dos distribuciones de probabilidad, la media de bits necesarios para identificar un evento de un conjunto de posibilidades, si un esquema de codificación está basado en una distribución de probabilidad dada q , más que en la verdadera distribución p . La entropía cruzada para dos distribuciones p y q sobre el mismo espacio de probabilidad se define como:

$$H(p, q) = E_p[-\log q] = H(p) + D_{KL}(p||q)$$

Dónde $H(p)$ es la entropía de p , y D_{KL} es la divergencia de KullBack-Leibler

La pérdida de entropía cruzada mide el rendimiento de un modelo de clasificación cuyo resultado es un valor de probabilidad entre 0 y 1. La pérdida de entropía cruzada aumenta a medida que la probabilidad prevista diverge de la etiqueta real

Los datos se han obtenido entrenando el modelo preentrenado de cada red utilizando las base de

datos de imágenes que hemos generado.

Viendo los resultados anteriores se ha decidido utilizar la red Alexnet sin ningún tipo de congelación en sus capas interiores ya que es la que ha obtenido mejores resultados para la predicción en la validación y el coste era mínimo.

3.4.1. Conclusiones y resumen del bloque de clasificación

Para cada video se han extraído y analizado aquellos frames en los que ha habido una o varias detecciones y que son candidatos a ser humo.

En el apartado 4 se expone todo el análisis de esta clasificación y se detallan tanto los aciertos como los fallos que se obtienen de la clasificación para todos los vídeos.

El bloque funciona tal y como se esperaba desde su definición, recibe un recorte de imagen candidato a humo y decide a que clase pertenece entre las clases de humo y no humo.

Se ha podido identificar la mayor parte de bloques de humo como humo y el número de falsas es relativamente pequeño

3.5. Analizador de salida del algoritmo:

Finalmente, el último bloque del sistema consiste en un analizador de los resultados del bloque de clasificación para obtener un resultado general de si hay humo o no en el vídeo.

Se han definido tres reglas, una para describir si en una imagen hay humo / no humo, otra para definir una etiqueta para cada segundo del video y finalmente la última para clasificar un conjunto de segundos del video, cada bloque encadenado al anterior.

A continuación se explica el detalle de su funcionamiento:

3.5.1. Análisis por imagen:

Cada imagen del video puede tener 0 o más detecciones en su interior. Siempre que no se haya hecho ninguna detección en la imagen, se etiquetará esa imagen como **no humo** en la posición del vector de imágenes. Si hay detecciones seguirá la siguiente prioridad:

- Si se clasifica algún elemento del interior de la imagen como humo, independientemente de cuantos bloques más haya, se etiquetará como humo.
- Si se detecta no humo, y no se ha detectado humo en la misma imagen, se etiquetará como no humo.

Éste análisis es crítico ya que es de máxima prioridad saber si se ha detectado humo

3.5.2. Clasificación única por segundo:

Una vez se ha etiquetado cada imagen, cada segundo del vídeo está compuesto por un conjunto de recortes N igual al ratio de imágenes por segundo con el que está codificado el video. Si una imagen se etiqueta como humo su valor será 1, sino será 0.

La etiqueta para esa imagen se definirá como el promedio entre las N etiquetas calculadas en el punto anterior y se aplicará un umbral en 0.5. Si el número es mayor a 0,5 se etiquetará como humo, si es menor como no humo. Si el número de humo es igual al de no humo detectado, la prioridad será del humo.

A continuación hay un ejemplo ilustrativo de un vídeo con un ratio de 5 imágenes por segundo en el que se aplica éste método:

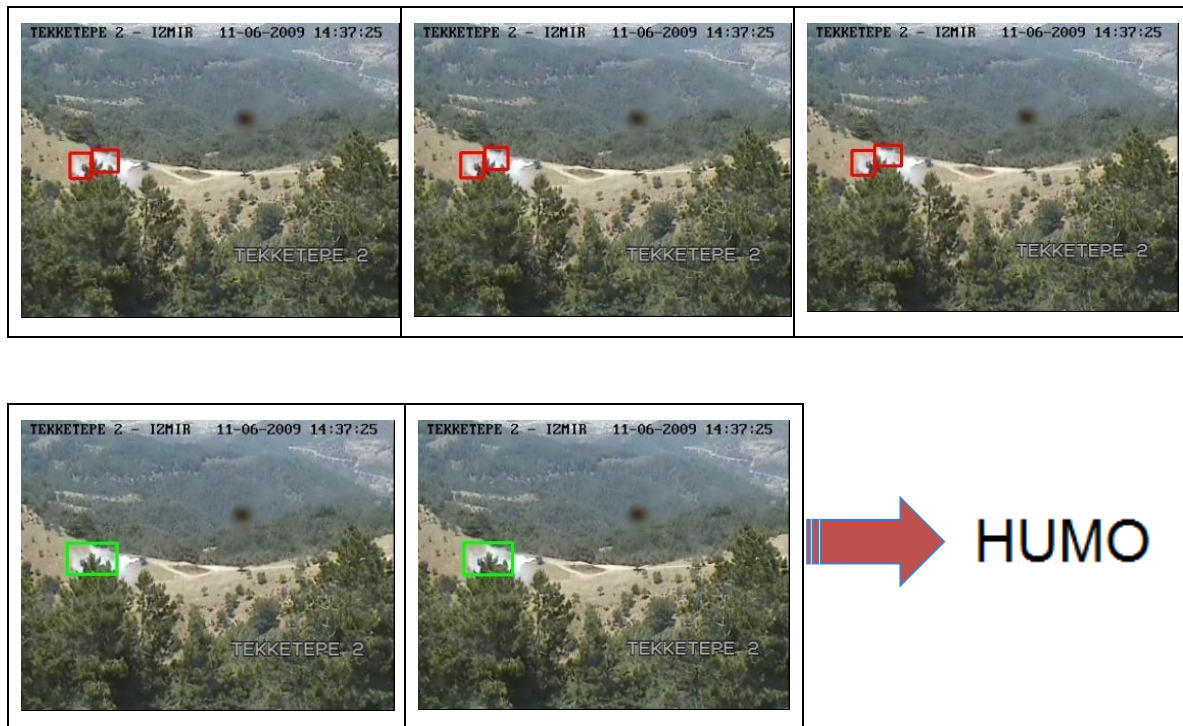


Figura 20. Ejemplo de la salida del bloque de clasificación por segundo. Promedia y calcula si hay mayoría de humo o no humo.

Con este procedimiento conseguimos que si se ha hecho una detección como humo entre varias detecciones no humo, esta se filtre y no genere una falsa alarma en el sistema.

3.5.3. Decisión final de humo /no humo en el vídeo

Finalmente se define éste último bloque de decisión.

Se toman las etiquetas obtenidas con el cálculo anterior y se busca promediar en un intervalo de tiempo más largo un conjunto de valores. Si una imagen se etiqueta como humo su valor será 1, sino será 0.

El humo es un fenómeno que en el momento que empieza a detectarse va creciendo y creciendo, con lo que no deja de detectarse hasta que se extingue. Aplicando esta teoría se define que, para cada conjunto de 10 segundos de video, se utiliza el valor obtenido en la clasificación por segundo y se promedian aplicando un umbral en el valor 0.5. Si el número es mayor a 0.5 se etiquetará como humo, si es menor como no humo. Si el número de humo es igual al de no humo detectado, la prioridad será del humo.

Este último filtro permite que si existe un pequeño intervalo de tiempo en el que la clasificación que se está haciendo es errónea, esto permita filtrar esta futura falsa alarma de nuestro sistema de detección de humo.

A continuación se muestra un ejemplo del esquema final para este conjunto de bloques:

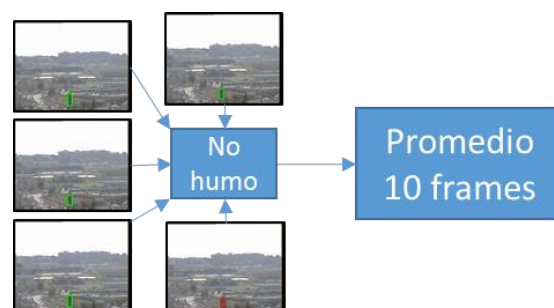


Figura 21. Ejemplo de la salida del bloque final del algoritmo. Indica para los últimos 10 segundos si ha habido humo o no humo

3.5.4. Conclusiones del analizador de salida del algoritmo:

Con este bloque se ha conseguido añadir un método que permite identificar si hay humo o no utilizando los resultados de la clasificación del bloque de redes neuronales.

Uno de los principales problemas que puede surgir en una aplicación que utilice este algoritmo para la clasificación de humo son las falsas alarmas. Contemplando los casos en los que se han generado falsas alarmas utilizando promedios por frame y por segundos del vídeo y teniendo en cuenta las características principales del humo en el tiempo desde que aparece el humo hasta que se va, este bloque permite eliminar de una manera eficiente las principales falsas alarmas conocidas. Éstas han sido principalmente dos:

- Existencia de recorte de una imagen clasificado como humo de forma aislada
- Conjunto de falsas alarmas que aparecen durante breves instantes entre frames en los que no existe humo.

4. Resultados

La detección de humo para todos los vídeos con humo ha sido prácticamente inmediata en cuanto se analizan frames en los que el humo está comenzando a aparecer ya que se ha adaptado el bloque de detección del fondo a las características del humo para así detectarlo correctamente.

Los resultados del bloque de detección de humo han sido los esperados ya que su objetivo, detectar nuevos bloques de humo para cada frame, se ha cumplido para todos los vídeos.

A continuación se explica el análisis de los resultados obtenidos del bloque de clasificación, que servirán para dar resultados globales del proyecto. Éstos resultados dependen directamente de la buena detección de los bloques de humo en las escenas y al cumplirse ésta condición se ha podido hacer el análisis

Por un lado se analiza el resultado de la clasificación de imágenes en sí misma para cada vídeo explicando los detalles y por otro lado los resultados globales obtenidos de la c, que aplica unas reglas a los resultados de la clasificación para que, siguiendo una lógica, las posibles falsas alarmas puedan ser mitigadas.

La única limitación que se ha observado del algoritmo para detectar el humo en las imágenes es debido a la velocidad de ejecución a la que se ejecuta la clasificación de cada blob por frame.

Ésta limitación ha surgido en el momento en que hay múltiples detecciones de elementos que se mueven en un frame. El proceso de tener que introducir múltiples imágenes (una por cada contorno detectado) por frame en el clasificador, hace que cada clasificación tenga que esperar a que se hayan clasificado todos los elementos para pasar a clasificar el siguiente frame.

La clasificación utilizando la red neuronal es costosa computacionalmente y en cuanto existen varios elementos detectados para su clasificación esto hace que los resultados que se obtienen de cada frame no sigan la velocidad a la que se ejecuta el vídeo. Por eso se ha planteado el análisis de los resultados a partir de la clasificación que se ha hecho de los blobs detectados y no de la clasificación a tiempo real.

4.1. Análisis y comentarios de los resultados de cada video de la base de datos:

Para cada vídeo analizado se muestran los casos más relevantes tanto en detección de humo como en falsas alarmas y se añade su matriz de confusión. Cada matriz de confusión se ha calculado manualmente haciendo un análisis de cada frame con detecciones que se han extraído de cada vídeo.

La suma de todos los elementos que aparecen en la matriz de confusión es el número de detecciones que se han hecho para los frames analizados.

Se busca obtener el mayor número de aciertos en la diagonal descendente y el mínimo en la ascendente, ya que para la diagonal descendente se contabilizan los elementos que eran humo y no humo que han sido clasificados correctamente como humo y no humo respectivamente, y para la ascendente se busca tener el mínimo, ya que son las falsas alarmas y los casos en los que el humo ha sido clasificado como no humo.

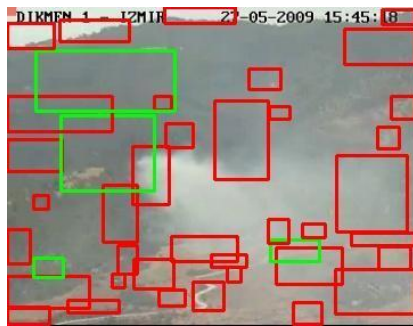
En alguno de los resultados se comentarán algunos casos de la clasificación que se hizo con la base de datos con menos imágenes de no humo, para ver la mejora que ha supuesto añadir más imágenes en el modelo.

El humo viene dado por los rectángulos en rojo y el no humo en color verde para las imágenes de los ejemplos siguientes.

Videos con humo:

1. CamiCota-2:

El video comienza ya con humo de fondo con lo que el sistema va detectando el posible humo nuevo. Para cada detección se ha decidido que es humo en todos los casos con lo que el acierto es del 100%.



En el frame 39 hay un movimiento brusco de la cámara y ya pierde todo el sentido seguir analizando ya que aparecen multitud de detecciones.



Figuras 22 y 23. Múltiples detecciones al moverse la imagen (izquierda).

Detección de humo correcta en cami-cota2.

Ésta fue la prueba para ver si debía poner la limitación al número de rectángulos a analizar, ya que si no se utilizaría mucho tiempo de cálculo para no conseguir resultados útiles.

De los bloques anteriores al movimiento brusco se ha obtenido la siguiente matriz de confusión:

	REAL	
	HUMO	NO HUMO
CamiCota-2		
HUMO	42	0
NO HUMO	0	0

Tabla 4. Matriz de confusión del video Camicota2.

Se observa pues una clasificación correcta del humo.

2. Cantera:

El video de Cantera es un video bastante estático y no se detecta nada en la imagen hasta que empieza a aparecer el humo.



Figura 24. Humo bien clasificado en Cantera

La matriz de confusión obtenida ha sido:

Cantera	REAL	
	HUMO	NO HUMO
HUMO	181	0
NO HUMO	0	0

Tabla 5. Matriz de confusión del vídeo Cantera.

La clasificación del humo en este caso ha sido muy acertada.

Para la antigua base de datos se detectó el siguiente bloque en el que se clasificaba humo como humo pero ahora ya se clasifica correctamente como humo



Figura 25. Blob de Humo detectado como no humo y dos blobs detectados correctamente como humo

3.Carena:

En el video de Carena se puede ver como se mueven las hojas de los árboles por el viento pero no las estamos detectando como posibles candidatos, que ya es lo que busca el algoritmo.

La matriz de confusión es la siguiente:

Carena	REAL	
	HUMO	NO HUMO
HUMO	89	0
NO HUMO	0	0

Tabla 6. Matriz de confusión del vídeo Carena.

Para este caso hasta que el humo no comienza a detectarse el sistema no encuentra ningún candidato a humo y cuando se encuentra se clasifican todos los candidatos como humo correctamente.

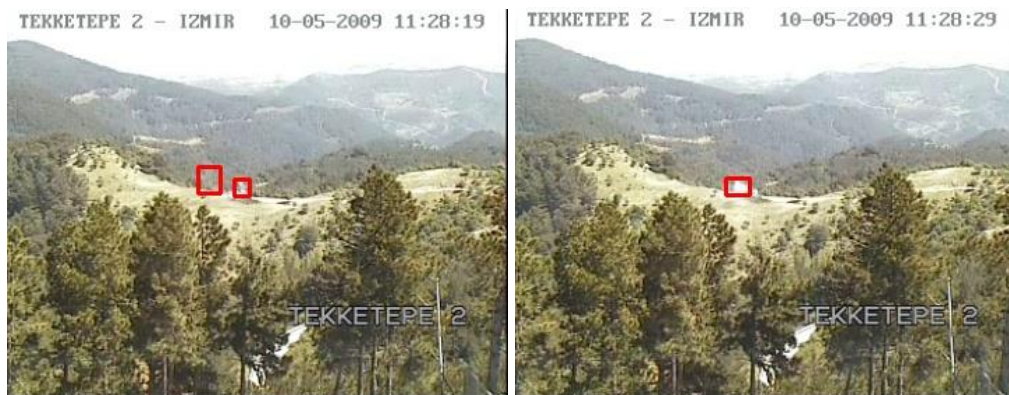


Figura 26. Dos blobs de Humo detectados como humo(1) y un blob de humo detectado correctamente como humo(2)

4. CarenaTaca:

En CarenaTaca el humo comienza a surgir lentamente desde detrás de unos árboles. La matriz de confusión obtenida es:

CarenaTaca	REAL	
	HUMO	NO HUMO
HUMO	381	0
NO HUMO	117	6

Tabla 7. Matriz de confusión del vídeo Carenataca

Se observa que no aparece ninguna falsa alarma en el sistema pero que la clasificación de varios bloques de humo se está clasificando erróneamente.

Utilizando la antigua base de datos se obtenía algunas falsas alarmas que se han corregido con el nuevo modelo actualizado.

Ejemplo comparación:



Figura 27. Blob de Humo y no humo clasificados erróneamente para un test anterior con menos imágenes en base de datos.



Figura28. Blobs de humo y no humo detectados correctamente

Imágenes clasificadas como no humo siendo humo:



Figura 29. Errores de clasificación de humo como no humo. Clasificación errónea (1) y clasificación errónea debido a que el recuadro en el que aparece el humo es muy grande y aparecen más elementos que no son humo.

En el primer caso sí que se trata de un fallo del clasificador dado que la proporción que tiene ese blob es de un tamaño similar a los de la base de datos y se considerará un error leve ya que anteriormente a este frame se había detectado correctamente el humo y el sistema podría detectar el humo.

Por otro lado, para el segundo caso el blob a analizar es mucho más grande y como la base de datos de humo está formada por imágenes que no contienen muchos más elementos a parte del humo, la clasificación la da como no humo, que en el contexto tiene mucho sentido ya que no se ha entrenado con este tipo de imágenes en las que aparece humo en una parte pequeña del blob.

Con lo que el resultado global para ésta imagen sería aceptable al haber podido detectar la gran mayoría del humo y no tener ninguna falsa alarma.

5. LLac-2:

En esta escena aparece un lago, una carretera y en el fondo comienza a aparecer humo lentamente. La matriz de confusión calculada es la siguiente:

Llac-2	REAL	
	HUMO	NO HUMO
HUMO	362	8
NO HUMO	0	94

Tabla 8. Matriz de confusión del vídeo Llac2.

Para este vídeo se habían encontrado 110 falsas alarmas de 438 frames revisados con la base de datos anterior, para la nueva el número de falsas alarmas se ha visto reducido drásticamente a 8 gracias a los nuevos valores del clasificador.

Detecciones de humo y no humo correctamente clasificados:



Figura 30. Blobs de no Humo detectado correctamente (1) Bloos de humo y no humo correctamente clasificados

Detección de no humo clasificado erróneamente como humo:



Figura 31. Falsas alarmas al clasificar elementos como humo siendo no humo. Frames aislados entre frames no humo.

En este caso han sido aisladas entre frames de no humo pero hay que tenerlas en cuenta para el cálculo global de errores en la clasificación.

6. Pont:

En Pont aparece un puente en el que a su lado comienza a surgir una columna de humo.

La matriz de confusión obtenida es :

Pont	REAL	
	HUMO	NO HUMO
HUMO	350	0
NO HUMO	1	16

Tabla 9. Matriz de confusión del vídeo Pont

De nuevo la detección del humo ha sido prácticamente perfecta y los elementos que no eran humo se han clasificado correctamente.

Elementos clasificados como no humo correctamente:



Figura 32. Blobs de no humo clasificados correctamente

En este caso se trata de los vehículos que avanzan por la carretera.

Se ha detectado un bloque de humo como no humo pero teniendo en cuenta que estaba entre blobs bien clasificados como humo, no afecta al buen resultado de la clasificación para éste vídeo.



Figura 33. Blob de humo clasificado como no humo

7. Playa

Este vídeo se ha añadido a la base de datos para comprobar que, en el caso que aparezca humo no proveniente de un incendio, se clasifica correctamente como humo.

La escena transcurre en una playa en la que un helicóptero desprende humo al volar.



Figura 34. Clasificación del helicóptero entero como no humo

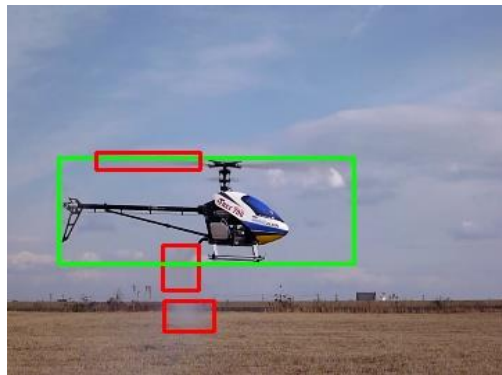
La matriz de confusión obtenida es la siguiente:

	REAL	
	HUMO	NO HUMO
playa		
HUMO	100	9
NO HUMO	6	130

Tabla 10. Matriz de confusión del vídeo Playa

Se puede ver pues de los resultados que se detecta correctamente el humo cuando comienza a aparecer y que en la mayoría de casos se clasifica correctamente los elementos que son no humo.

Sin embargo aparecen 9 falsas alarmas producidas por las hélices del avión:



Estas aparecen por la forma que tienen las hélices al girar tan rápido y por el fondo de nubes que se encuentra detrás de estas.

La forma de las hélices rotando recuerda a la forma en la que tiene el humo y es lo que hace que la clasificación no sea la correcta.

Figura 35. Humo generado por el avión clasificado correctamente como humo. Hélices del avión en movimiento como humo erróneamente.

Vídeos sin humo:

En estos vídeos aparecen diferentes escenas para comprobar si se generarían falsas alarmas en vídeos fuera del ámbito forestal.

1. Cam-car

Este escena transcurre en una rotonda en la que gira una furgoneta a su alrededor.

La matriz de confusión obtenida ha sido :

Cam-car	REAL	
	HUMO	NO HUMO
HUMO	0	11
NO HUMO	0	147

Tabla 11. Matriz de confusión del vídeo Cam-car.

La gran mayoría de detecciones se han clasificado correctamente y aparecen 11 falsas alarmas, éstas son en su mayoría cuando acaba de girar la furgoneta en que la puerta trasera tiene un tono parecido al del color del humo.

Ejemplo detección y clasificación correcta de no humo:



Figura 36. Blobs de no humo clasificados correctamente

Falsas alarma para la escena:



Figura 37. Algunos blobs clasificados correctamente como no humo y falsa alarma generada por la puerta.

2. Camino

En éste fragmento se analiza el movimiento de un hombre cruzando por un campo de olivos.

La matriz de confusión ha sido la siguiente:

	REAL	
	HUMO	NO HUMO
Camino		
HUMO	0	46
NO HUMO	0	123

Tabla 12. Matriz de confusión del vídeo Camino.

Para este caso el número de falsas alarmas ha sido más elevado que para el caso anterior. Se ha comprobado que para este vídeo y para "Fondo- casa" los humanos que aparecen en la imagen se están considerando en algunas ocasiones como humo.

Que se consideren como humo parece ser un problema, pero éste tipo de vídeos con la cámara tan cerca del objetivo no es precisamente el caso de análisis de este proyecto, pero sirve para reconocer puntos en los que se puede trabajar en un futuro para poder aplicar esta clasificación de humo en nuevas situaciones.

Ejemplo no humo detectado incorrectamente:



Figura 38. Humano clasificado erróneamente como hum

3. Contorno

Para esta escena se analiza el comportamiento de un vehículo teledirigido sobre un cielo que va variando con el tiempo y por tanto la iluminación de la imagen.

La matriz que se ha obtenido es la siguiente:

Contorno	REAL	
	HUMO	NO HUMO
HUMO	0	0
NO HUMO	0	103

Tabla 13. Matriz de confusión del video Contorno.

Todos los blobs detectados se han clasificado correctamente como no humo. *Ejemplo:*



Figura 39. Blobs de objeto en movimiento correctamente clasificado como no humo

Se observa pues que aunque varíe la iluminación de la imagen se ha hecho correctamente la detección y clasificación de estos blobs.

4. Fondo casa

En éste vídeo tal y como se ha comentado, anteriormente, aparece una persona, que en este caso se dirige a una piscina.

La matriz de confusión que se ha obtenido ha sido la siguiente:

Fondo-casa	REAL	
	HUMO	NO HUMO
HUMO	0	36
NO HUMO	0	167

Tabla 14. Matriz de confusión del vídeo Fondo Casa.

En el vídeo se puede observar que dado cierto instante, la clasificación comienza a generar falsas alarmas cuando se detecta a la persona.



Figura 40. Blobs de persona clasificado incorrectamente. Vídeo sin estar en el contexto de paisajes.

Esto supone un problema ya que como hemos comprobado, las personas en los vídeos analizados están siendo clasificadas de forma errónea.

Para éstos casos especiales en que aparecen personas relativamente cerca de la cámara, se podría plantear, en caso que se quisiera aplicar este clasificador en otro tipo de prototipos o situaciones, de crear una nueva base de datos añadiendo personas en movimiento y que en lugar de definirse dos clases se definiesen tres: humo, seres vivos/personas y resto, en la salida de la red neuronal.

5. Paisaje

Ésta escena es la más escueta de la base de datos ya que sólo se aprecia movimiento en un momento del vídeo y luego en el resto no se hace ninguna detección a clasificar

Su matriz de confusión ha sido:

paisaje	REAL	
	HUMO	NO HUMO
HUMO	0	0
NO HUMO	0	15

Tabla 15. Matriz de confusión del vídeo Paisaje.

Se han clasificado correctamente todos los blobs detectados.

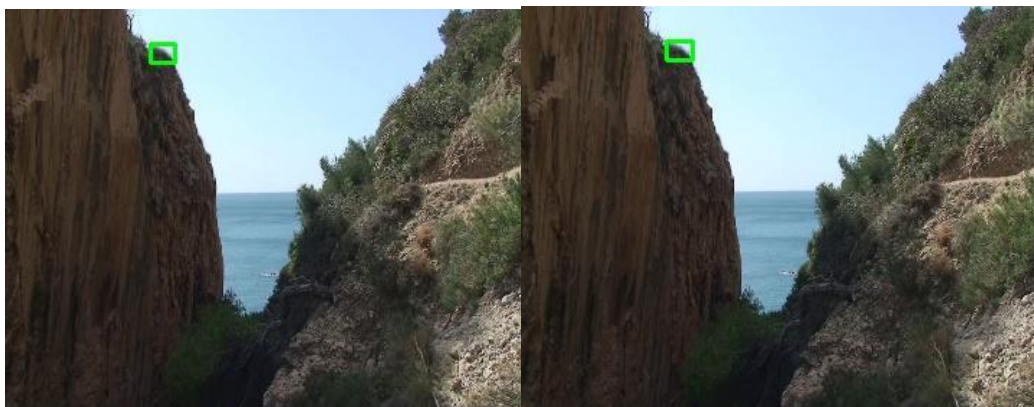


Figura 41. Blobs de no humo detectados correctamente

6. Paisajes

Este vídeo contiene algunos paisajes en los que hay movimiento



Figura 42. Blobs detectados totalmente después de un cambio de imagen en la cámara

Estas dos imágenes anteriores son de la transición de una imagen a otra con lo que se ha detectado como blob a analizar toda la imagen.

La matriz de confusión obtenida ha sido la siguiente:

paisajes	REAL	
	HUMO	NO HUMO
HUMO	0	77
NO HUMO	0	340

Tabla 16. Matriz de confusión del vídeo Paisajes.

Se han generado varias falsas alarmas, algunas provocadas por la transición de un paisaje a otro y otras que se analizan a continuación



Figura 43. Blob de agua rompiendo en la orilla clasificado incorrectamente como humo(1) Blobs de nubes clasificados incorrectamente como humo

En los ejemplos se observa que tanto algún movimiento de las nubes como el movimiento de las olas al chocar contra la playa generan falsas alarmas. Estos casos son importantes ya que las

formas de las nubes y de las olas al chocar son bastante parecidos a la estructura del humo, motivo por el que posiblemente la clasificación no es correcta para estos dos casos.

La clasificación de la gran mayoría de blobs ha sido correcta pero ha fallado en algunos casos como los dos anteriores.

7. Verde col

En éste último vídeo se analiza como entra por la izquierda de la imagen un objeto y se va detectando. Por otro lado hay movimiento de nubes en la imagen.

La matriz de confusión ha sido la siguiente:

Verde-col	REAL	
	HUMO	NO HUMO
HUMO	0	8
NO HUMO	0	221

Tabla 17. Matriz de confusión del vídeo VerdeCol.

La detección de no humo ha sido muy buena en general exceptuando 8 falsas alarmas.

Éstas se han generado en frames aislados unos de otros entre frames que se han clasificado correctamente como no humo.



Figura 44. Blobs aislados detectados como no humo

El primer bloque se ha detectado en la nube en un instante y se ha clasificado como no humo.

El resto de casos son como la segunda imagen en la que aparece el objeto pero se clasifica como humo durante menos de 1 frame y luego se vuelve a clasificar correctamente como no humo.

Ejemplo de blobs clasificados como no humo correctamente:



Figura 45. Blobs clasificados correctamente como no humo

4.2. Análisis de los resultados finales

La gran mayoría del humo que se ha detectado en los vídeos se ha conseguido clasificar correctamente.

Las falsas alarmas detectadas suceden en momentos en los que, o bien se lleva detectando humo desde hace varios frames, o bien son detecciones puntuales en los que, para varios frames anteriores y posteriores, no hay detección de humo. También se ha detectado algún grupo de falsas alarmas que aparecen en pequeños grupos de frames (que en realidad corresponderían a 1 o 2 segundos de vídeo), motivo por el cual se ha planteado crear el bloque de análisis para en una aplicación real poder eliminar éstas falsas alarmas.

Para obtener un resultado cuantitativo de la clasificación se ha generado una matriz de confusión para los casos que hay humo, otra para los que no lo hay y una para tener una visión global:

· Matriz de confusión global de humo:

	REAL	
	HUMO	NO HUMO
HUMO	1505	17
NO HUMO	124	246

Tabla 18. Matriz de confusión del total de vídeos con humo.

· Matriz de confusión global de no humo:

	REAL	
	HUMO	NO HUMO
HUMO	0	178
NO HUMO	0	1116

Tabla 19. Matriz de confusión del total de vídeo sin humo.

· Matriz de confusión global de toda la clasificación:

	REAL	
	HUMO	NO HUMO
HUMO	1505	195
NO HUMO	124	1362

Tabla 20. Matriz de confusión del total de vídeos del proyecto

A continuación hay una tabla resumen de los resultados de la clasificación partiendo de las matrices anteriores

	Frames analizados	Detecciones	% Acierto humo	% Falsas alarmas
Vídeos Humo	1456	1892	88,22%	6,46%
Vídeos No Humo	881	1294	-	13,53%
Todos Vídeos	2337	3186	88,22%	12,5%

Tabla 21. Tabla de resultados finales de clasificación

El % acierto de humo se calcula como el humo detectado correctamente entre la suma de los detectados correctamente y el humo detectado como no humo.

Observando la tabla anterior vemos que el % Acierto de humo para los vídeos con humo ha sido del 88,22 % con lo que podemos hablar de una clasificación bastante acertada y el % de falsas alarmas respecto al número de blobs que son no humo ha sido del 6%.

Por otro lado se observa que para los vídeos que no aparece humo y que son de varios tipos las falsas alarmas aumentan al 13,53 % (que era lo esperado ya que no son vídeos puramente dentro del contexto).

Finalmente si se hace un análisis global del algoritmo, el % de acierto del humo sería del 88,22 % y para las falsas alarmas se queda un 12,5%. Se observa por tanto que a medida que nos alejamos de vídeos relacionados con el entorno forestal, el % de falsas alarmas tiende a crecer.

Se puede afirmar pues que, para estos valores de acierto, el algoritmo ha cumplido en gran medida con el principal objetivo del proyecto.

Como conclusión, para poder aplicar este algoritmo en una aplicación en la vida real, sería necesario utilizar el bloque de análisis a la salida del clasificador para que algunas de las falsas alarmas que se han encontrado fuesen excluidas.

A continuación se muestran los resultados para la salida utilizando el analizador de videos. La salida da un valor para cada 10 segundos de humo(H) o no humo(NH)

Nombre Video	Tiempo en que aparece humo	Resultado
CamiCota-2.avi	Desde el inicio	En cuanto detecta nuevo humo al fondo, se empieza a clasificar como humo
CarenaTaca.avi	38s - fin	Se clasifica como humo a partir del segundo 50.
Pont.avi	35s – 3m	No humo hasta el segundo 40 que ya empieza a detectar el humo
Cantera.avi	1m 33s– fin	Se empieza a clasificar como humo a partir del 1'40
Llac-2.avi	5s – 3m 15s	Comienza a clasificarse como humo en el segundo 10

Cam-car.avi	-	Se clasifica como no humo para todo el tiempo excepto durante un período de 10 segundo en los que se ha clasificado como humo. En el resto se ha clasificado correctamente
Camino.avi	-	Se clasifica como no humo en todo instante de tiempo
Contorno.avi	-	Se clasifica como no humo en todo instante de tiempo
Fondo-casa.avi	-	Se clasifica como no humo para todo el tiempo excepto durante un período de 10 segundo en los que se ha clasificado como humo. En el resto se ha clasificado correctamente
Paisaje.avi	-	Se clasifica como no humo en todo instante de tiempo
Paisajes.avi	-	Se clasifica como no humo en todo instante de tiempo
Playa.avi	20s - fin	Se detecta correctamente el humo de la imagen con lo que se clasifica como humo a partir del segundo 20
Verde-col.avi	-	Se clasifica como no humo en todo instante de tiempo

Tabla 22. Tabla de resultados finales del analizador de vídeos

Observando los resultados vemos que para la mayor parte en la que ha habido humo éste se ha detectado correctamente y solo ha habido una falsa alarma para los casos en que se han clasificado personas cercanas a la cámara como humo. Con lo que se podría considerar que el algoritmo funciona y clasifica correctamente para la gran mayoría de casos y cuando se podría generar una falsa alarma, el bloque de análisis conseguiría mitigarla para la mayoría de casos en una aplicación real. Cabe remarcar que mientras se está adaptando el humo al modelo, esos segundos de inicio no se detectan como candidatos a humo y por eso vemos que no es inmediata la detección y clasificación del humo en los vídeos.

5. Budget

Este proyecto no consiste en ningún prototipo de producto para el mercado ya que se trata de una colaboración entre la Universidad Politécnica de Cataluña y ADTelecom.

A continuación se detalla el coste teórico que tendría este proyecto si se contratase un ingeniero Junior fijando un coste por hora estándar.

El número de horas que aparecen en la tabla de a continuación es una estimación entre horas de desarrollo + horas de aprendizaje + horas de búsqueda de información

Horas	Semanas	Horas totales	Precio Hora
20 / semana	20	400	8 €

No existe ningún tipo de cargo económico por el software utilizado ya que Python , openCV y pytorch son totalmente gratuitos al ser open source products.

Con lo que finalmente el coste del proyecto sería de un total de 3200€ suponiendo que se pagasen todas las horas y no solo las de desarrollo

6. Conclusiones y mejoras a futuro

Este proyecto ha conseguido cumplir los resultados esperados a la hora de detectar y clasificar el humo.

Es cierto que la idea inicial del proyecto estaba enfocada a hacer un sistema a tiempo real para la detección del humo, pero con la inclusión del bloque de redes neuronales para la clasificación de imagen esto ha influido en un cambio de objetivo, conseguir una buena clasificación de los elementos nuevos que aparecen en los vídeos.

La clasificación en algunas ocasiones no ha sido la correcta como era de esperar y eso generaría falsas alarmas en un sistema real, pero por otro lado el número de falsas alarmas es mucho menor al número de elementos que no son humo que se han detectado correctamente como no humo, con lo que estos podrían ser tratados como casos aislados en la clasificación en la mayoría de casos. Por estos casos se ha incluido el planteamiento del último bloque para poder corregir la gran mayoría de las falsas alarmas que podría tener una aplicación real.

Cabe decir que si se hubieran invertidos más horas al proyecto los resultados se podrían haber mejorado y se podrían eliminar muchas falsas alarmas mejorando el clasificador.

De cara a futuras mejoras de los bloques del proyecto, la mejora principal consistiría en obtener una base de datos mucho más grande para la red neuronal. El impacto que tiene tener más muestras con las que se ha entrenado el sistema permite tomar mejores decisiones a la hora de clasificar una imagen como humo o no humo.

También, el haber podido conseguir muchas más imágenes para la base de datos implicaría que en lugar de utilizar una red preentrenada hubiera sido posible entrenar la red completamente desde 0 con todas las imágenes, y se podría comprobar si esto mejoraría los resultados.

Por otro lado también se debería trabajar con más vídeos y con más variedad de situaciones. Esto permitiría ver nuevas problemáticas que puedan existir con la clasificación que se hace actualmente. Inicialmente se esperaba obtener vídeos de la empresa que colaboró en 2011 con el anterior proyecto de detección de humo, pero no han podido proporcionar nuevos vídeos para esta ocasión, con lo que el número de videos útiles ha sido limitado.

Finalmente y como última mejora sería muy interesante comprobar si existe algún tipo de red neuronal que no hayamos considerado en este proyecto que se adapte mejor a la clasificación que se busca hacer y en ese caso ponerla en práctica.

Bibliography:

Lista de referencias utilizadas:

- [1] Zivkovic, Z. (2004, August). Improved adaptive Gaussian mixture model for background subtraction. In Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on (Vol. 2, pp. 28-31). IEEE.
- [2] Zivkovic, Z., & Van Der Heijden, F. (2006). Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern recognition letters*, 27(7), 773-780. Zhang, Q. X., Lin, G. H.,
- [3] Stauffer, C., & Grimson, W. E. L. (1999, June). Adaptive background mixture models for real-time tracking. In *cvpr* (p. 2246). IEEE.
- [4] Tao, C., Zhang, J., & Wang, P. (2016, December). Smoke detection based on deep convolutional neural networks. In 2016 International conference on industrial informatics-computing technology, intelligent technology, industrial information integration (ICIICII) (pp. 150-153). IEEE.
- [5] Çelik, T., Özkaramanlı, H., & Demirel, H. (2007, September). Fire and smoke detection without sensors: Image processing based approach. In *Signal Processing Conference, 2007 15th European* (pp. 1794-1798). IEEE.
- [6] Piccinini, P., Calderara, S., & Cucchiara, R. (2008, October). Reliable smoke detection in the domains of image energy and color. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on* (pp. 1376-1379). IEEE.
- [7] Zhang, Y. M., Xu, G., & Wang, J. J. (2018). Wildland Forest Fire Smoke Detection Based on Faster R-CNN using Synthetic Smoke Images. *Procedia engineering*, 211, 441-446.
- [8] Yin, M., Lang, C., Li, Z., Feng, S., & Wang, T. (2018). Recurrent convolutional network for video-based smoke detection. *Multimedia Tools and Applications*, 1-20.
- [9] Labati, R. D., Genovese, A., Piuri, V., & Scotti, F. (2013). Wildfire smoke detection using computational intelligence techniques enhanced with synthetic smoke plume generation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(4), 1003-1012.
- [10] Vipin, V. (2012). Image processing based forest fire detection. *International Journal of Emerging Technology and Advanced Engineering*, 2(2), 87-95.
- [11] Chunyu, Y., Jun, F., Jinjun, W., & Yongming, Z. (2010). Video fire smoke detection using motion and color features. *Fire technology*, 46(3), 651-663.
- [12] Han, X. F., Jin, J. S., Wang, M. J., Jiang, W., Gao, L., & Xiao, L. P. (2017). Video fire detection based on gaussian mixture model and multi-color features. *Signal, Image and Video Processing*, 11(8), 1419-1425.

[13] Surit, S., & Chatwiriya, W. (2011, May). Forest fire smoke detection in video based on digital image processing approach with static and dynamic characteristic analysis. In *Computers, Networks, Systems and Industrial Engineering (CNSI), 2011 First ACIS/JNU International Conference on* (pp. 35-39). IEEE.

[14] Yu, C., Mei, Z., & Zhang, X. (2013). A real-time video fire flame and smoke detection algorithm. *Procedia Engineering*, 62, 891-898.

[15] Zamal, M. F. B., Sayed, S., Bhuiyan, T., & Rahman, M. (2017). An Efficient Multi-sensing and GSM Equipped Fire Monitoring System. In *MATEC Web of Conferences* (Vol. 140, p. 01003). EDP Sciences.